



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

Programa de Pós-Graduação em Matemática

Mestrado Profissional - PROFMAT/CCT/UFCG



PROFMAT

Carlos Gustavo Barreto de Farias Júnior

Funções discretas e algoritmos de programação na robótica com Arduino

Campina Grande - PB

agosto/2025



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

Programa de Pós-Graduação em Matemática

Mestrado Profissional - PROFMAT/CCT/UFCG



PROFMAT

Carlos Gustavo Barreto de Farias Júnior

Funções discretas e algoritmos de programação na robótica com Arduino

Trabalho de Conclusão de Curso apresentado ao Corpo Docente do Programa de Pós-Graduação em Matemática - CCT - UFCG, na modalidade Mestrado Profissional, como requisito parcial para obtenção do título de Mestre.

Orientador: Dr. Marcelo Carvalho Ferreira

Campina Grande - PB
agosto/2025

Universidade Federal de Campina Grande - UFCG
Sistema de Bibliotecas - SISTEMOTECA
Catalogação de Publicação na Fonte. UFCG - Biblioteca Central

F224f Farias Júnior, Carlos Gustavo Barreto de.
Funções discretas e algoritmos de programação na robótica com
Arduino / Carlos Gustavo Barreto de Farias Júnior. – 2025.
76 f. : il. color.

Dissertação (mestrado em Matemática) – Universidade Federal de
Campina Grande, Centro de Ciências e Tecnologia, 2025.

“Orientação: Prof. Dr. Marcelo Carvalho Ferreira”.

Referências.

1. Funções Discretas. 2. Algoritmos de Programação. 3. Robótica
com Arduino. 4. Cibernética Matemática. I. Ferreira, Marcelo Carvalho.
II. Título.

UFCG/BC

CDU 519.714(043.3)


FICHA CATALOGráfICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

Carlos Gustavo Barreto de Farias Júnior


Funções discretas e algoritmos de programação na robótica com Arduino

Trabalho de Conclusão de Curso apresentado ao Corpo Docente do Programa de Pós-Graduação em Matemática - CCT - UFCG, na modalidade Mestrado Profissional, como requisito parcial para obtenção do título de Mestre.


Trabalho aprovado. Campina Grande - PB, 29 de agosto de 2025:

Documento assinado digitalmente
 **MARCELO CARVALHO FERREIRA**
Data: 02/10/2025 10:47:32-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Marcelo Carvalho Ferreira
Orientador

Documento assinado digitalmente
 **NATAN DE ASSIS LIMA**
Data: 02/10/2025 14:09:57-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Natan de Assis Lima
Membro externo - UEPB

Documento assinado digitalmente
 **LUIZ ANTONIO DA SILVA MEDEIROS**
Data: 03/10/2025 08:56:18-0300
Verifique em <https://validar.iti.gov.br>

**Prof. Dr. Luiz Antônio da Silva
Medeiros**
Membro interno - UFCG

Campina Grande - PB
agosto/2025

Dedico este trabalho à minha família, que foi o meu alicerce e motivação.

Agradecimentos

Agradeço imensamente à minha esposa, que me deu um grande suporte ao longo de todo o curso e cuidou de tudo durante o curso de verão, quando precisei passar muito tempo distante. Meu imenso obrigado! A meus pais, que me incentivaram desde cedo a estudar e buscar conhecimento. Fizeram de tudo para que eu trilhasse uma boa vereda e, agora, só tenho a agradecer. À minha avó Socorro, que desde cedo foi muito atenciosa para comigo, e que deu as primeiras perspectivas do que era ser professor, quando me levava para seu escritório na secretaria de educação da nossa cidade, onde pude ter acesso a muitas revistas e materiais didáticos, o que estimulou bastante meu interesse pelo estudo. Agradeço especialmente a meus professores, Marcelo, Rodrigo, Leomiques, Romildo, Daniel e Deise que, em certos momentos onde o desânimo chegava, me incentivaram. Acima de tudo, agradeço pelo ensinamento que me deram. Meu eterno agradecimento! A todos os funcionários do UAMAT. Foi uma grande satisfação poder conhecer a cada um! Obrigado por tudo!

“A Matemática é a mais simples, a mais perfeita e a mais antiga de todas as ciências.” (Jacques Hadamard)

Resumo

O presente trabalho traz uma sequência didática com aplicações de funções discretas e de algoritmos de programação na robótica e mecatrônica da plataforma Arduino, para ser aplicada em turmas do Ensino Médio do primeiro ao terceiro ano. A elaboração deste recurso tem como objetivo principal mostrar ao estudante como os conteúdos de funções e sequências estudados no Ensino Médio têm uma aplicação na programação de dispositivos eletrônicos, além de mostrar como a área da robótica é bastante matematizada. Um outro objetivo, não menos importante, é contemplar a segunda competência da BNCC, que diz respeito a desenvolver o pensamento científico crítico e criativo para engendrar soluções e resolver problemas. Competência na qual, segundo o ranking do PISA 2022, o Brasil ficou na 44^a posição, dentre 64 países participantes. Por fim, almeja-se que o uso da robótica possa tornar as aulas mais interessantes para o estudante.

Palavras-chave: Funções discretas. Algoritmos de Programação. Arduino.

Abstract

The present work introduces a didactic sequence with applications of discrete functions and programming algorithms in robotics and mechatronics on the Arduino platform, designed for use in classes from the 10th to the 12th grade. The main objective of this resource is to demonstrate to students how the concepts of functions and sequences studied in high school have an application in the programming of electronic devices, as well as to highlight how the area of robotics is quite mathematized. An equally important goal is to address the second general competence of the BNCC, which concerns developing critical and creative scientific thinking for problem-solving and solution development. This is particularly relevant considering Brazil's 44th position among 64 countries in the PISA 2022 ranking in this area. Finally, we aim that the use of robotics could make the classes more interesting to the student.

Keywords: Discrete functions. Programming Algorithms. Arduino.

Lista de ilustrações

Figura 1 – Fonte: Tchê(página do Facebook), 2025	15
Figura 2 – Fonte: Quelle antiquarisch und privat, 2025. Domínio público.	16
Figura 3 – Fonte: site do clube Olhar Digital. Acesso em 05/08/2025.	17
Figura 4 – Fonte: Site Stylo Urbano, Acesso em 06/08/2025.	19
Figura 5 – Fonte: Livro <i>Jacquard weaving and designing</i> , p.124	19
Figura 6 – Fonte: Site Wikiwand. Acesso em 06/08/2025	20
Figura 7 – Fonte: site do G1, 2025	21
Figura 8 – Fonte: site da Irmandade da Santa Casa de Santos, 2025	22
Figura 9 – Fonte: Autor.	28
Figura 10 – Fonte: Autor.	30
Figura 11 – Fonte: site Christian Cechinel, 2025.	34
Figura 12 – Fonte: Reddit, 2025	36
Figura 13 – Fonte: eBay, 2025.	38
Figura 14 – Fonte: Autor. Feito com auxílio do software Arduino IDE.	40
Figura 15 – short text	41
Figura 16 – Fonte: Site da Amazon.	42
Figura 17 – Fonte: BNCC, p. 531,(2025).	44
Figura 18 – Fonte: BNCC, p. 533,(2025).	44
Figura 19 – Fonte: autor, a partir da BNCC, p. 534,(2025).	44
Figura 20 – Fonte: Apostila Robótica Livre com Arduino - p. 14	49
Figura 21 – Fonte: Apostila Robótica Livre com Arduino - p. 14	49
Figura 22 – Fonte: Apostila Robótica livre com Arduino, p. 15	50
Figura 23 – Fonte: Apostila Robótica livre com Arduino, p. 7.	51
Figura 24 – Fonte: Apostila Robótica livre com Arduino, p. 7.	52
Figura 25 – Fonte: autor	52
Figura 26 – Fonte: Autor.	54
Figura 27 – Fonte: Apostila Robótica livre com Arduino, p.13.	54
Figura 28 – Fonte: Autor.	55
Figura 29 – Fonte: Apostila Robótica livre com Arduino, p. 20	56
Figura 30 – Fonte: Autor. Feito com auxílio do software Arduino IDE.	57
Figura 31 – Fonte: Autor. Feito com auxílio do software Paint.	58
Figura 32 – Fonte: Autor. Adaptada da apostila robótica livre com Arduino	59
Figura 33 – Fonte: Autor. Adaptação da apostila robótica livre com Arduino	60
Figura 34 – Fonte: Apostila Robótica livre com Arduino, p. 16.	61
Figura 35 – Fonte: Autor. Adaptado da apostila Robótica livre com Arduino.	62

Figura 36 – Fonte: apostila Robótica livre com Arduino, p. 17.	63
Figura 37 – Fonte: Descrições: Autor. Código: Autor, adaptado da apostila Robótica livre com Arduino, p. 18.	63
Figura 38 – Fonte: apostila Robótica livre com Arduino, p. 23.	65
Figura 39 – Fonte: Autor. Adaptado a partir da Apostila Robótica livre com Arduino.	66
Figura 40 – Fonte: apostila Robótica livre com Arduino, p. 20.	67
Figura 41 – Fonte: Autor. Adaptado a partir da apostila Robótica livre com Arduino, ps. 20 e 21.	68
Figura 42 – Fonte: autor, foto cedida por integrante do grupo desenvolvedor.	70
Figura 44 – Fonte: Foto cedida por estudante	71
Figura 43 – Fonte: autor, foto cedida por integrante do grupo desenvolvedor.	71

Sumário

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.2	Organização	14
2	HISTÓRIA	15
2.1	A automação na antiguidade	15
2.2	As revoluções industriais	18
2.3	A automação na atualidade	20
3	CONHECIMENTOS MATEMÁTICOS RECOMENDADOS PARA A APLICAÇÃO DA SEQUÊNCIA DIDÁTICA	23
3.1	Conjuntos	23
3.1.1	Relações entre conjuntos	24
3.1.2	Operações com conjuntos	24
3.1.3	Conjuntos numéricos	26
3.1.4	Conjunto Discreto	28
3.2	Funções	29
3.3	Conversão de bases numéricas	32
4	O ARDUINO E A PROGRAMAÇÃO	34
4.1	Algoritmos de programação	34
4.2	O Arduino	36
4.2.1	Composição	37
4.2.2	As portas	37
5	SEQUÊNCIA DIDÁTICA	43
5.1	Aplicação	43
5.2	Sequência didática	46
5.2.1	Primeiro encontro: Introdução ao Arduino e ao multímetro	47
5.2.2	Segundo encontro: Medição de resistência elétrica	48
5.2.3	Terceiro encontro: Função da porta digital e o Pisca com LED	50
5.2.4	Quarto encontro: Controlar LED com sinal PWM	53
5.2.5	Quinto encontro: Controlar um pisca de LED com potenciômetro	55
5.2.6	Sexto encontro: Sensor de temperatura	58
5.2.7	Sétimo encontro: Sensor de luminosidade e sensor ultrassônico	61

5.2.8	Oitavo encontro: Buzzer e sensor de campo magnético	65
5.2.9	Análise da aplicação	69
6	CONCLUSÃO	73
	REFERÊNCIAS	75

1 Introdução

A prototipagem é o processo de transformar uma ideia em um produto testável. O conceito de prototipagem pode variar, dependendo do contexto. Um protótipo pode ser um programa de computador, um dispositivo eletrônico ou mecânico, uma peça de vestuário, ou um brinquedo, por exemplo. No geral, um protótipo é uma versão inicial de um produto que pode ser testada. Após os testes, podem ser observados problemas e possíveis melhorias a serem feitas. [19]

No contexto educacional, a prototipagem é parte do processo de *design thinking*, que consiste em 5 etapas, que são:

- *1.Empatia*: Conhecer bem as pessoas a quem se quer ajudar e, através disso saber quais seus problemas.
- *2.Definição*: A partir do conhecimento das pessoas, deve-se definir o problema a ser resolvido para ajudá-las.
- *3.Ideação*: Definido o problema, o estudante deve criar várias ideias e métodos diferentes pelos quais ele acredita que possa resolvê-lo.
- *4.Prototipação ou prototipagem*: Desenhar e escrever soluções e, a partir disso, criar um protótipo o mais próximo possível da realidade.
- *5.Testagem*: Definir um público com o qual será feita a testagem do protótipo. Nela, deve-se verificar problemas, voltando às fases anteriores quando necessário.

Após a testagem e a resolução dos problemas observados, o protótipo pode ser produzido em larga escala para ser distribuído para o público alvo. Porém é necessário lembrar que o processo de testagem não está finalizado. É necessário sempre estar atento à experiência do usuário e fazer novas adequações criando um produto cada vez melhor. [20]

Neste trabalho, utilizaremos como ferramenta de aplicação o kit Arduino Uno com alguns sensores e componentes, que é um conjunto integrado de prototipagem de dispositivos eletrônicos. Ele consiste em duas partes: um programa de computador, que é utilizado para escrever os códigos de comando dos protótipos, nos quais serão implementadas funções matemáticas e algoritmos de programação, e uma parte física que são as placas, fios, resistores, sensores e saídas, com os quais, montamos os dispositivos.

Este trabalho objetiva a construção de uma sequência didática utilizando esquemas de montagem e programação com Arduino da apostila *Robótica livre com Arduino*, criada pela Secretaria de educação do Estado de Pernambuco com o apoio de uma equipe técnica. Nas atividades a serem propostas, serão enfatizados os aspectos matemáticos dos dispositivos e das programações, de forma que o estudante possa aplicar seu conhecimento de funções e algoritmos de programação na robótica com o Arduino.

Dentre os tópicos que serão abordados, alguns temas de matemática são as funções discretas e termos relacionados, modelagem matemática por meio das funções e desigualdades e os algoritmos de programação. O conceito de função será, inicialmente, aplicado no estudo da comunicação de portas digitais, PWM¹ e analógicas², que interpretam níveis de tensão elétrica como números naturais, os quais serão transformados e utilizados para se obter a informação desejada, seja temperatura, campo magnético, distância, ou intensidade luminosa. É justamente nestas transformações que o conceito de função discreta é empregado. A programação envolvendo os modelos matemáticos é feita com uma linguagem variante da C++ utilizada na programação do Arduino, onde as representações matemáticas são adaptadas a esta linguagem. Dentre os dispositivos a serem acoplados aos protótipos, vale destacar os sensores de temperatura, campo magnético, distância e luminosidade, além das saídas de som e luz. Tais variáveis serão adequadamente medidas e controladas pela manipulação das funções discretas utilizadas na programação. Por exemplo: O sensor de luminosidade pode ser acoplado a uma saída que liga uma lâmpada, a qual mantém o ambiente iluminado. Mas o nível de luminosidade suficientemente baixo ou alto têm de ser analisados com a utilização do sensor, que recebe a luminosidade ambiente e a transforma em valores numéricos, os quais ele imprime no monitor do computador. Com tais valores, a programação pode ser ajustada.

1.1 Objetivos

Objetivos gerais

Este trabalho propõe uma maneira de ver a abstrata ideia de função, apresentada ao estudante no Ensino Médio, aplicada num contexto operacional e mais palpável para o(a) estudante, relacionando as funções com elementos da natureza, como espaço, temperatura, campo magnético e luz. Além disso, almejamos mostrar como a matemática é uma ferramenta importante na área de tecnologia e desenvolvimento de facilidades,

¹ PWM significa *Pulse Width Modulation*. As portas PWM são pinos que simulam uma saída analógica. Isto significa que podem emitir uma tensão de saída modulada em vários níveis (256, no máximo) que vão de 0 a 5v para controlar um dispositivo.

² Portas analógicas são pinos que modulam e interpretam uma tensão de 0 a 5V em 1024 níveis distintos.

enfatizando como ela pode ser crucial na missão de construir um mundo mais justo, e equalitário, melhorando a qualidade de vida das pessoas por meio da solução de problemas utilizando a automação.

Para alcançar tais objetivos, esquematizamos o trabalho em objetivos específicos.

Objetivos específicos

- Apresentar partes da história das automações partindo da antiguidade até alguns autômatos modernos e linguagens de programação.
- Abordar o conceito de função e sua representação na linguagem de programação Arduino em diversos códigos básicos, mostrando como tais conhecimentos possibilitam a inventividade e a solução de problemas através da prototipagem com o Arduino;
- Exibir as funções e os algoritmos e linguagens de programação como recursos da matemática para a melhor operação e compreensão dos protótipos robóticos de automação, que garantem ao(a) estudante mais facilidade nos processos inventivos e na criação das soluções ótimas, instigando neles a criatividade.

1.2 Organização

Este trabalho se estrutura em quatro capítulos, dos quais, o capítulo 2, apresenta o surgimento e desenvolvimento das automações desde a antiguidade, passando pela revolução industrial e algumas automações modernas, incluindo a consequente criação das linguagens de programação.

No Capítulo 3, são introduzidos os conhecimentos matemáticos necessários e suficientes para o professor que deseja aplicar a sequência didática que será apresentada no Capítulo 5.

Em seguida, no Capítulo 4, são apresentados os conceitos de algoritmo de programação e das linguagens de programação, enveredando o conteúdo para a linguagem do Arduino, que será a linguagem utilizada na sequência didática.

A sequência didática será apresentada no Capítulo 5, onde tratamos dos pré-requisitos básicos para a sua aplicação, tempo e os materiais necessários, além dos planos de cada encontro, em sequência, com os esquemas de montagem, códigos de programação e exercícios a serem propostos para os estudantes.

2 História

2.1 A automação na antiguidade

A origem da automação remonta a tempos antigos, desde quando o ser humano percebeu a necessidade de facilitar seus trabalhos e garantir a sustentabilidade de uma população. Em buscas arqueológicas na região onde se situava o Egito Antigo, foram encontradas evidências de sistemas de irrigação baseados em comportas, que facilitavam o trabalho de irrigar as plantações. Outros povos antigos, como os gregos e romanos, utilizavam mecanismos automáticos para controlar moinhos e portas.

As clepsidras: relógios de água da antiguidade

As clepsidras eram relógios de água que consistiam num recipiente com um pequeno furo no fundo e feito com um material de um tipo que se pudesse ver o nível de água em seu interior. Existem achados de clepsidras feitas com cobre, cerâmica e, até mesmo, vidro. As clepsidras foram utilizadas nas cortes gregas e romanas para controlar o tempo de fala dos advogados [2], evitando assim discursos demasiadamente prolongados. Estes artefatos marcavam o tempo a partir do nível de água restante nele. A água escoava pelo pequeno buraco no fundo de um clepsidra para um outro depósito que a armazenava e, desta forma, a partir de marcações feitas na superfície da clepsidra, a conseqüente diminuição no volume de água era utilizada para determinar o número de horas transcorridas. A figura 1 ilustra um tipo de clepsidra.

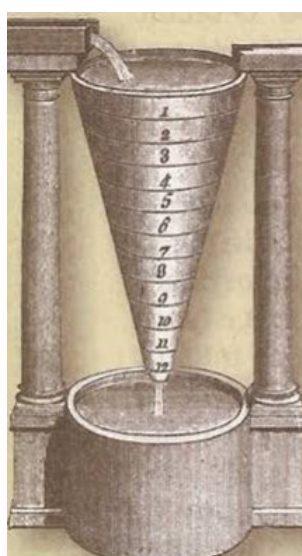


Figura 1 – Fonte: Tchê(página do Facebook), 2025

A princípio, algumas clepsidras eram feitas com paredes verticais. Porém, como as marcações não funcionavam corretamente, devido ao fluxo de água cada vez mais lento, alguns construtores de clepsidras começaram a fabricar algumas com paredes não verticais, mas que formavam um ângulo de 70° com a horizontal, compensando a diminuição do fluxo de água com um volume de água cada vez menor para cada marcação, como está ilustrado na figura 1. Apesar disto, as marcações ainda não funcionavam de maneira perfeita. Uma investigação melhor à respeito deste problema, trouxe a solução de utilizar formas que, futuramente viriam a ser conhecidas como paraboloides de revolução. Um desenho de 1903 (figura 2) mostra o califa Harune Arraxide, presenteando o sacro imperador romano-germânico Carlos Magno com uma clepsidra cuja forma se assemelha a um parabolóide de revolução.



Figura 2 – Fonte: Quelle antiquarisch und privat, 2025. Domínio público.

Apenas por volta do séc. XVII, com os estudos do cálculo diferencial, foi possível obter a demonstração de que a forma adequada para o recipiente com água deveria ser um parabolóide de revolução pois, desta forma, o fluxo de água se torna constante. [2]

A clepsidra mais antiga que o ser humano já teve acesso foi encontrada na tumba de *Ahmenotep*, a qual foi datada de 1524 a.C. [13] e existem registros de que clepsidras

foram utilizadas por chineses, babilônios, indianos, egípcios e gregos.

Tal invenção só se tornou obsoleta no séc. XIV, quando foi inventado o relógio mecânico.

Heron de Alexandria

No séc. I d.C., um notável inventor grego, conhecido como Heron de Alexandria, escreveu sobre óptica, pneumática, máquinas de guerra, autômatos e geometria.

Em seu livro "Mecânica", ele definiu como máquinas simples cinco dispositivos que são utilizados para mover pesos em uma certa trajetória: a roda, a polia, a alavanca, a cunha e o parafuso [18].

Para cada uma dessas máquinas simples, ele define do que se tratam, ensina a construção de cada uma delas e mostra diversas formas de utilizá-las. Porém, nenhuma dessas máquinas podem ser considerada um autômato, pois todas elas, a princípio, dependem da força humana para serem utilizadas.

Porém, uma invenção que podia gerar força mecânica sem depender de força humana ou animal foi a eolípila de Heron (figura 3), considerada por muitos historiadores como a primeira máquina a vapor da história.



Figura 3 – Fonte: site do clube Olhar Digital. Acesso em 05/08/2025.

A eolípila era uma máquina térmica, ou seja, transformava calor em trabalho mecânico. Contudo, a eolípila não chegou a ser utilizada para produzir grandes quantidades de trabalho mecânico e sua aplicação é, até hoje, desconhecida. Os registros dizem

que o uso de máquinas a vapor só veio a acontecer no séc. XVIII, quando as primeiras máquinas a vapor foram utilizadas nos setores industriais e automotivos. [8]

2.2 As revoluções industriais

No séc. XVIII, na Europa ocidental, iniciou-se uma expansão industrial impulsionada pela necessidade enxergada pelos empresários e burgueses europeus de gerar um crescimento na economia, na produção e na produtividade. Para tal, foram favorecidos pela grande oferta de mão de obra e o acesso a máquinas dado pelo invento das máquinas a vapor e o desenvolvimento das automações industriais. A título de exemplo, inicialmente, os setores agrário e artesanal tiveram um grande aumento de produtividade, o que gerou uma perspectiva de crescimento em forma de metas de expansão de mercado e de acúmulo de capital e meios de produção. A indústria têxtil iniciou tal revolução, com a introdução de máquinas como o tear mecânico, o descaroçador de algodão, o Tear de Jacquard, a lançadeira, a fiadeira automática e a máquina a vapor rotativa de James Watt [12]

O tear de Jacquard

Dentre as grandes invenções da época, uma que teve uma característica interessante em relação ao que será tratado no trabalho foi o tear de Jacquard. O tear de Jacquard foi o primeiro tear mecânico já inventado. Joseph Marie Jacquard era filho de tecelões e se formou como um mecânico e comerciante [4]. Durante seu convívio com seu pai, notara a dificuldade que residia na atividade da tecelagem, na qual eram necessárias duas pessoas para o trabalho. O próprio Jacquard disse que seu pai levava uma semana para bordar dez centímetros⁰ de tecido. [7]

Jacquard desenvolveu um método para tecer formas e desenhos nos tecidos, que consistia em utilizar uma tira de papelão com furos milimetricamente recortados que, quando passavam pelo tear, determinavam como os fios ficariam entrelaçados dando destaque a alguns pontos, enquanto que outros ficavam lisos. Os pontos da faixa de papelão com furos deixavam o pontos destacados, enquanto que os pontos sem furo deixavam a peça lisa. Desta maneira, ele formavam desenhos, os quais serviriam de padrão para a máquina de tecelagem seguir e, neste padrão, tecer a peça com desenhos. Essa tira de papelão e seus desenhos em forma de buracos milimetricamente recortados funcionava como um código binário, pois o tecer variava de duas formas distintas: liso ou destacado.

Sem perda de generalidade, podemos considerar que o espaço furado seria o 0, enquanto que o espaço não furado seria o 1. Desta forma, os desenhos feitos na faixa

⁰ Na citação acima, entenda-se que na tecelagem de uma peça de tecido, a peça é retangular e a largura da peça é fixa, a depender da estrutura do tear e que, ao tecer, a peça vai ganhando comprimento. Assim, o autor quis dizer 10 centímetros de comprimento a mais na peça.

de papelão podem ser escritos como códigos binários. O código de desenho em papelão se tornou tão peculiar que todo tear mecânico que usava o código no papelão era dito um tear de Jacquard.

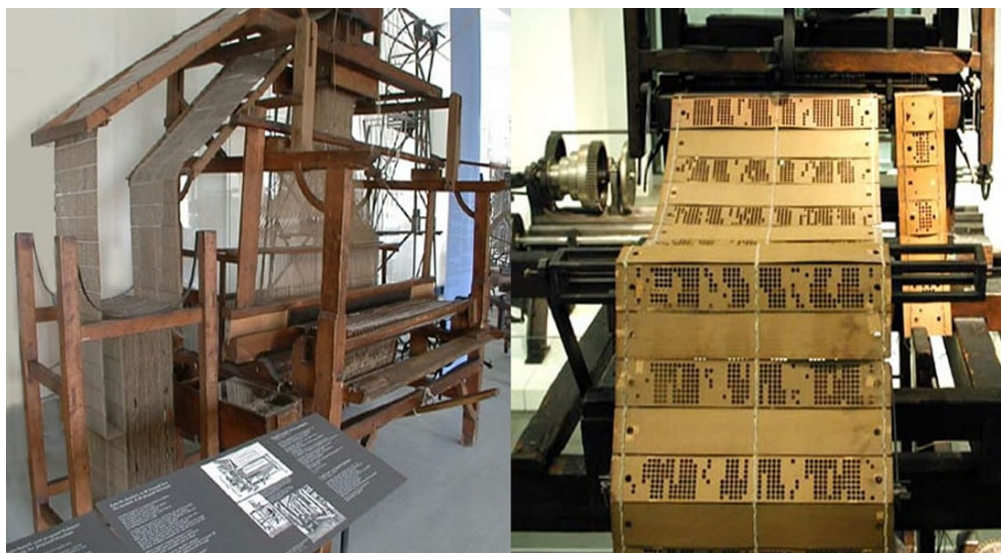


Figura 4 – Fonte: Site Stylo Urbano, Acesso em 06/08/2025.

Uma característica interessante dos desenhos nos tecidos feitos em teares Jacquard é que, caso se vire o tecido para o lado avesso, veremos a negativa desse desenho, ou seja cada ponto que está destacado em um lado, no outro fica liso, e vice-versa. Por tal criação, o tear de Jacquard é considerado por alguns como o primeiro computador já criado. [7]

Na figura 5, veremos um exemplo de código Jacquard.



Figura 5 – Fonte: Livro *Jacquard weaving and designing*, p.124

Governador centrífugo de James Watt

Com o desenvolvimento da indústria, os processos industriais puderam se tornar cada vez mais complexos. Inicialmente, as máquinas tinham seu ritmo de trabalho

e funcionamento controlados por técnicos treinados. Porém, com o tempo, os maquinários começaram a se tornar cada vez mais sofisticados, de forma que tornou-se necessária a invenção de mecanismos de controle automáticos, que conseguissem controlar o ritmo de trabalho das máquinas, liberando o homem para que pudesse se ocupar em trabalhos mais rebuscados. Foi neste contexto que James Watt desenvolveu o governador centrífugo: um mecanismo de controle do ritmo de trabalho de máquinas que se baseava no volume de vapor que podia ser armazenado em seus cilindros.

A figura 6, apresenta uma fotografia de um governador centrífugo acoplado a um motor a vapor.



Figura 6 – Fonte: Site Wikiwand. Acesso em 06/08/2025

O volume de vapor que cabia nos seus cilindros servia como limitador para o ritmo de trabalho da máquina. Desta forma, o governador centrífugo de James Watt configurou-se como uma das mais marcantes automações já criadas na história.

2.3 A automação na atualidade

A automação e a robótica são, certamente, pontos altos dos dias atuais. Isto pode ser vislumbrado nos mais variados meios, como na medicina, astronomia, transportes, segurança e educação, onde muitos trabalhos e inventos foram realizados, facilitando ainda mais a vida das pessoas. Atualmente, pode-se ter acesso a linguagens de programação muito mais sofisticadas e de escrita mais fácil. Do início do século XX até hoje,

os problemas envolvendo automação ganharam cada vez mais relevância, dada a sua crescente aplicação e importância cada vez maior no dia a dia.

Além da automação do trabalho físico humano, vemos também a automatização do pensamento, onde a inteligência artificial toma a cena, demonstrando grandes feitos, alguns dos quais seriam impossíveis à mente humana comum. Por exemplo, um estudo feito por pesquisadores da Universidade de Tel Aviv e da Universidade de Ariel, ambas em Israel, produziu um código fonte de inteligência artificial que foi capaz de traduzir a língua acadiana contida em tabuletas de argila milenares, com detalhes sobre a vida, a sociedade, a economia, a religião e cultura dos babilônios por volta de 2000 a 3000 a.C.. De acordo com os pesquisadores, tais traduções não seriam possíveis sem o uso da inteligência artificial. [11] Em outro caso, no interior do Amazonas, em postos de saúde sobrecarregados, os farmacêuticos estão utilizando uma inteligência artificial para evitar erros graves em prescrições médicas. Esta tecnologia foi desenvolvida pela organização brasileira sem fins lucrativos NoHarm, com apoio da Google e da Amazon. [6]

Nos setores de montagem das indústrias, os robôs e automações reduzem os erros de produção, retrabalhos, custos com energia e mão de obra, além de aumentar bastante a produção.

Na exploração espacial, as automações são utilizadas com grande frequência, pois os artefatos robóticos resistem a ambientes que são altamente hostis ao ser humano, possibilitando o estudo e a exploração até mesmo de outros planetas, como, o planeta Marte. Na figura 7, segue uma ilustração computadorizada do robô opportunity, que foi enviado pela NASA a marte, e cuja missão findou no ano 2019, após 15 anos de estudo.

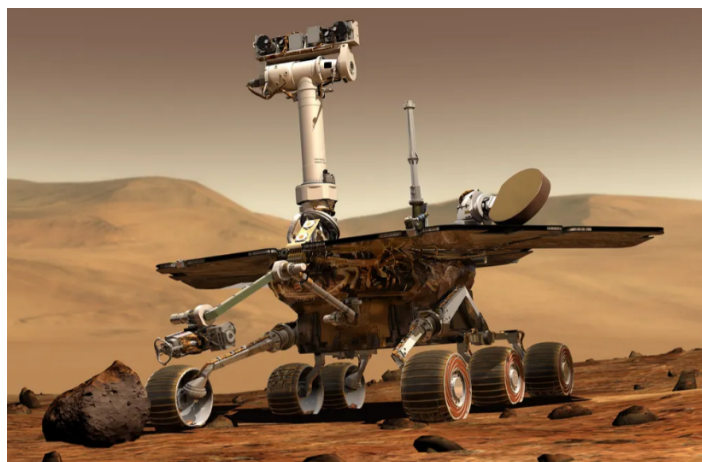


Figura 7 – Fonte: site do G1, 2025

No campo da medicina, um ponto a ser destacado é a cirurgia robótica. Esta modalidade entra na categoria dos procedimentos minimamente invasivos. Braços robóticos com pinças e prolongamentos extremamente eficientes são acoplados a um carro que

acompanha o paciente para fazer a cirurgia. O médico habilitado para tal tipo de procedimento controla o robô com um *Joystick* e, desta forma, realiza a cirurgia evitando erros e problemas como tremores, nervosismo e, além disso, tem uma melhor ergonomia, pois realiza toda a cirurgia sentado, o que é uma melhoria para as condições de trabalho dos médicos, como mostra a figura 8. Tal tipo de procedimento é utilizado em tratamentos de cânceres como o de ovário, por exemplo.



Figura 8 – Fonte: site da Irmandade da Santa Casa de Santos, 2025

3 Conhecimentos matemáticos recomendados para a aplicação da sequência didática

3.1 Conjuntos

O conceito de conjunto é primitivo. Um conjunto nada mais é do que uma coleção de elementos. Um conjunto pode ser caracterizado por meio de uma característica comum e exclusiva de seus objetos, representada por uma propriedade que define seus elementos. Contudo, podemos também ter conjuntos nos quais os elementos não têm propriedade em comum e são apenas representados entre as chaves($\{\}$).

Exemplo 1. *Considere o conjunto I dos números ímpares. Podemos representá-lo como*

$$I = \{3, 5, 7, 9, 11, 13, 15, \dots\}.$$

Dados um objeto x e um conjunto A , existem apenas duas possibilidades:

- (i) x é elemento de A . Portanto, escrevemos $x \in A$. (lê-se x pertence a A)
- (ii) x não é elemento de A . Logo, escrevemos $x \notin A$. (lê-se x não pertence a A)

Exemplo 2. *Seja P o conjunto dos números pares. Temos $2 \in P$ e $3 \notin P$.*

Exemplo 3. *Seja M o conjunto das raízes da equação $x(x + 1)(x - 3) = 0$. Neste caso, $-1 \in M, 2 \notin M$.*

Conjunto vazio

O conjunto vazio é aquele que não possui elementos. Seu símbolo representativo é \emptyset , mas também pode ser representado por $\{\}$.

Exemplo 4. *O conjunto A das soluções reais da equação $x^2 + 5 = 0$ é vazio. De fato,*

$$x^2 + 5 = 0 \iff x^2 = -5.$$

Como não existem números reais de quadrado negativo, a equação $x^2 + 5 = 0$ não possui soluções reais. Assim,

$$A = \emptyset.$$

3.1.1 Relações entre conjuntos

Duas relações fundamentais entre conjuntos são as relações de inclusão e de igualdade.

Relação de inclusão

Definição 3.1. Dizemos que um conjunto A está contido em um conjunto B , ou A é um subconjunto de B se todo elemento de A pertence a B . Se A está contido em B , denotamos tal relação por

$$A \subset B.$$

Caso A não esteja contido em B , escrevemos

$$A \not\subset B.$$

Para mostrar que $A \not\subset B$, basta exibir um elemento de A que não pertence a B .

Exemplo 5. Seja D o conjunto dos divisores de 60, $A = \{2, 3, 4, 5, 6, 7\}$ e $B = \{2, 3, 6\}$. Assim, tem-se

$$A \not\subset D, \text{ pois } 7 \in A \text{ e } 7 \notin D, \text{ e } B \subset D.$$

Relação de igualdade

Definição 3.2. Dois conjuntos A e B são ditos iguais se têm os mesmos elementos. Representamos tal relação por

$$A = B.$$

Observe que

$$A = B \iff A \subset B \text{ e } B \subset A.$$

3.1.2 Operações com conjuntos

Existem três operações fundamentais entre conjuntos, a saber, a união, a interseção e a complementação.

União de conjuntos

Definição 3.3. Dados dois conjuntos, A e B , a união de A e B é o conjunto de todos os elementos que pertencem a A ou a B . A notação da união de A e B é $A \cup B$.

Assim,

$$x \in A \cup B \iff x \in A \text{ ou } x \in B.$$

Exemplo 6. Sejam $P = \{1, 2, 3, 5, 7\}$ e $Q = \{4, 5, 6, 7, 8\}$, tem-se

$$P \cup Q = \{1, 2, 3, 4, 5, 6, 7, 8\}.$$

Intersecção entre conjuntos

Definição 3.4. Dados dois conjuntos A e B , a intersecção de A e B é o conjunto dos elementos que pertencem a A e também a B . Para representar esta operação utilizamos a notação $A \cap B$.

Em resumo,

$$x \in A \cap B \iff x \in A \text{ e } x \in B.$$

Exemplo 7. Sejam $P = \{1, 2, 3, 5, 7\}$ e $Q = \{4, 5, 6, 7, 8\}$, tem-se

$$P \cap Q = \{5, 7\}.$$

Complementar de um conjunto

Fixemos um conjunto U , denominado de *conjunto-universo*. Uma vez fixado o conjunto U , todos os elementos a serem considerados pertencem a U e todos os conjuntos a serem considerados são subconjuntos de U . [17]

Definição 3.5. Dado um conjunto A , o complementar de A , que representamos por A^c , é o conjunto de todos os elementos de U não pertencentes a A .

Definição 3.6. Dados dois conjuntos A e B , com $A, B \subset U$ o complementar de A em B é o conjunto dos elementos que pertencem a B e não pertencem a A . Para denotar o complementar de A em B , utilizamos a notação $B - A$. Observe que

$$B - A = B \cap A^c.$$

Exemplo 8. $P = \{1, 2, 3, 5, 7\}$ e $Q = \{4, 5, 6, 7, 8\}$, tem-se:

$$P - Q = \{1, 2, 3\} \quad \text{e} \quad Q - P = \{4, 6, 8\}.$$

3.1.3 Conjuntos numéricos

Números naturais (\mathbb{N})

O mais simples dos conjuntos numéricos é o conjunto dos *números naturais*, simbolizado pela letra \mathbb{N} . Os números naturais são os números utilizados para a operação de contagem ou para enumeração dos termos de uma sequência. Ou seja,

$$\mathbb{N} = \{1, 2, 3, 4, 5, 6, 7, \dots\}.$$

Números inteiros (\mathbb{Z})

Nem sempre, com $a, b \in \mathbb{N}$, temos $a - b \in \mathbb{N}$. Por exemplo, $3 - 5 \notin \mathbb{N}$. Para dar sentido a tais operações, introduzimos um conjunto, denominado conjunto dos números inteiros e representado pelo símbolo \mathbb{Z} . Este símbolo tem origem na letra inicial da palavra *zahlen*, que em alemão significa “números”.

Os *números inteiros* consistem dos números naturais, do zero e dos simétricos dos números naturais.

Portanto,

$$\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, \dots\}.$$

Números racionais (\mathbb{Q})

Um *número racional* é uma razão entre números inteiros. Representamos este conjunto por

$$\mathbb{Q} = \left\{ \frac{m}{n}; m, n \in \mathbb{Z}, n \neq 0 \right\}.$$

Todo número inteiro também pode ser representado por uma razão entre números inteiros. Desta forma, temos $\mathbb{Z} \subset \mathbb{Q}$. Os números racionais incluem os números inteiros e, pelo algoritmo da divisão euclidiana, os números decimais finitos e os infinitos que possuem período, também conhecidos como *dízimas periódicas*.

Alguns números não podem ser escritos como uma dízima periódica ou uma fração ordinária e, por isso, não pertencem a \mathbb{Q} . Tais números são ditos *irracionais*.

Definição 3.7. *Um número natural é dito primo se tem exatamente dois divisores.*

Proposição 3.1. *Se p é primo, então \sqrt{p} é irracional.*

Demonstração

Demonstração. Suponha, por absurdo, que, com p sendo um número primo,

$$\sqrt{p} \in \mathbb{Q}.$$

Com isto,

$$\sqrt{p} = \frac{m}{n}, \text{ com } m, n \in \mathbb{Z}, \text{ e } n \neq 0.$$

Como

$$(\sqrt{p})^2 = \left(\frac{m}{n}\right)^2,$$

obtemos

$$p = \frac{m^2}{n^2}.$$

Mas então $pn^2 = m^2$. Absurdo, pois o primeiro membro da equação tem um número ímpar de fatores iguais a p , enquanto que o segundo membro tem um número par de fatores iguais a p .

Desta maneira, demonstramos que, para todo número primo p ,

$$\sqrt{p} \notin \mathbb{Q}.$$

■

Números reais (\mathbb{R})

Um *número real* é um número que é racional ou irracional. O conjunto dos números reais, representado pelo símbolo \mathbb{R} , é então formado pelos números racionais unidos com aqueles que não são racionais.

$$\mathbb{R} = \mathbb{Q} \cup \mathbb{Q}^c.$$

Uma forma de representar os números reais consiste em considerar uma reta e escolher dois pontos distintos O e P para representarem os números 0 e 1, respectivamente. Por definição, o segmento OP mede 1 unidade de comprimento (1 u.c.) e é dito *segmento unitário*. O ponto O , ao qual chamaremos de origem, divide a reta em duas semirretas. A que contém P chamar-se-á a *semirreta positiva*. A outra será a *semirreta negativa*. Consideraremos que os pontos da semirreta positiva *estão à direita* da origem e os pontos da semirreta negativa *estão à esquerda* de O .

Seja X um ponto qualquer da reta \overleftrightarrow{OP} . Se o comprimento de OX é x , dizemos que x é a *abscissa* de X , caso X esteja à direita de O . Em caso contrário, dizemos que $-x$ é a *abscissa* de X .

O procedimento acima define uma correspondência biunívoca entre a reta \overleftrightarrow{OP} e o conjunto \mathbb{R} . Neste caso, a reta \overleftrightarrow{OP} é denominada *reta numerada* ou *eixo real*.

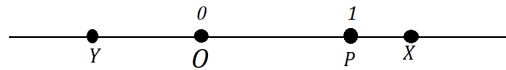


Figura 9 – Fonte: Autor.

Definição 3.8. Um conjunto é dito um intervalo real se é constituído por todos os números reais entre dois valores extremos $a, b \in \mathbb{R}$, $a < b$, dados, podendo incluir ou não estes extremos. Com os dois extremos, temos quatro possibilidades:

- i) O intervalo inclui a , mas não inclui b , daí é representado por $[a, b)$;
- ii) O intervalo não inclui a , mas inclui b , daí é representado por $(a, b]$;
- iii) O intervalo inclui tanto a , quanto b , daí é representado como $[a, b]$;
- iv) O intervalo não inclui nem a , nem b , daí, é representado por (a, b) .

Temos também os intervalos com apenas um ou nenhum extremo. Estes podem ser de cinco maneiras distintas:

- i) $(-\infty, a)$ Semirreta à esquerda de a , não contendo a ;
- ii) $(-\infty, a]$ Semirreta à esquerda de a , não contendo a ;
- iii) $(a, +\infty)$ Semirreta à direita de a , não contendo a ;
- iv) $[a, +\infty)$ Semirreta à direita de a , contendo a .
- v) $(-\infty, +\infty)$, compreendendo a totalidade dos números reais.

3.1.4 Conjunto Discreto

Dentre os conjuntos numéricos, é importante destacarmos a definição de conjunto discreto.

Definição 3.9. Sejam $X \subset \mathbb{R}$ e $a \in \mathbb{R}$. Dizemos que a é ponto de acumulação de X se, para todo $\epsilon > 0$, o intervalo $(a - \epsilon, a + \epsilon)$ contém algum elemento de X diferente do próprio a . Se $a \in X$ não é ponto de acumulação de X , então dizemos que a é um ponto isolado de X . [16]

Definição 3.10. Um conjunto é dito discreto se todos os seus elementos são pontos isolados dele, isto é, o conjunto não tem pontos de acumulação.

Exemplo 9. Os conjuntos, \mathbb{N} e \mathbb{Z} são conjuntos discretos.

De fato, dados $n \in \mathbb{Z}$ e $\epsilon = \frac{1}{2}$, o intervalo $(n - \frac{1}{2}, n + \frac{1}{2})$ não contém nenhum elemento pertencente a \mathbb{Z} distinto de n . Portanto, todo $n \in \mathbb{Z}$ é ponto isolado de \mathbb{Z} . Analogamente, todo $n \in \mathbb{N}$, é ponto isolado de \mathbb{N} .

Exemplo 10. O conjunto $A = \{0\} \cup \{\frac{1}{n}; n \in \mathbb{N}\}$ não é discreto.

De fato, $\forall \epsilon > 0$, tomando $n \in \mathbb{N}, n > \frac{1}{\epsilon}$, obtemos $\frac{1}{n} < \epsilon$. Logo, $\frac{1}{n} \in (-\epsilon, \epsilon)$. Assim, 0 é ponto de acumulação de A , ou seja, A não é discreto.

3.2 Funções

Desde o primeiro ano do Ensino Médio, é apresentada ao estudante a noção de função. As funções podem ser descritas da seguinte forma: um número dito *valor de entrada*, é associado a um segundo número, denominado *valor de saída*. O conjunto dos valores de entrada é chamado de *domínio*, enquanto que o conjunto ao qual pertencem os valores de saída é chamado de *contradomínio*. Esta ideia de valor de entrada e valor de saída estará bastante presente nos problemas e atividades que constam na sequência didática deste trabalho.

Nas atividades da sequência didática apresentada no Capítulo 5, são propostos exercícios nos quais o(a) estudante é instigado(a) a determinar funções para modelar os comandos das programações.

Assim sendo, dizemos que

Definição 3.11. Dados dois conjuntos A e B , dizemos que f é uma função de A em B , o que denotamos por $f : A \rightarrow B$, se f associa, por meio de uma regra, fórmula, algoritmo, etc., denominada sua lei de correspondência, a cada elemento de A um único elemento de B . Nesse caso, dizemos que A é o domínio e B o contradomínio da função f .

Dada uma função $f : A \rightarrow B$ e $x \in A$, o valor $f(x)$ é denominado de *imagem* de x por f . Dizemos que $f(x)$ é o *valor dependente* de x e que x é o *valor independente*.

Exemplo 11. Observe as correspondências por meio de diagramas de flechas entre os elementos dos conjuntos indicadas na figura 8:

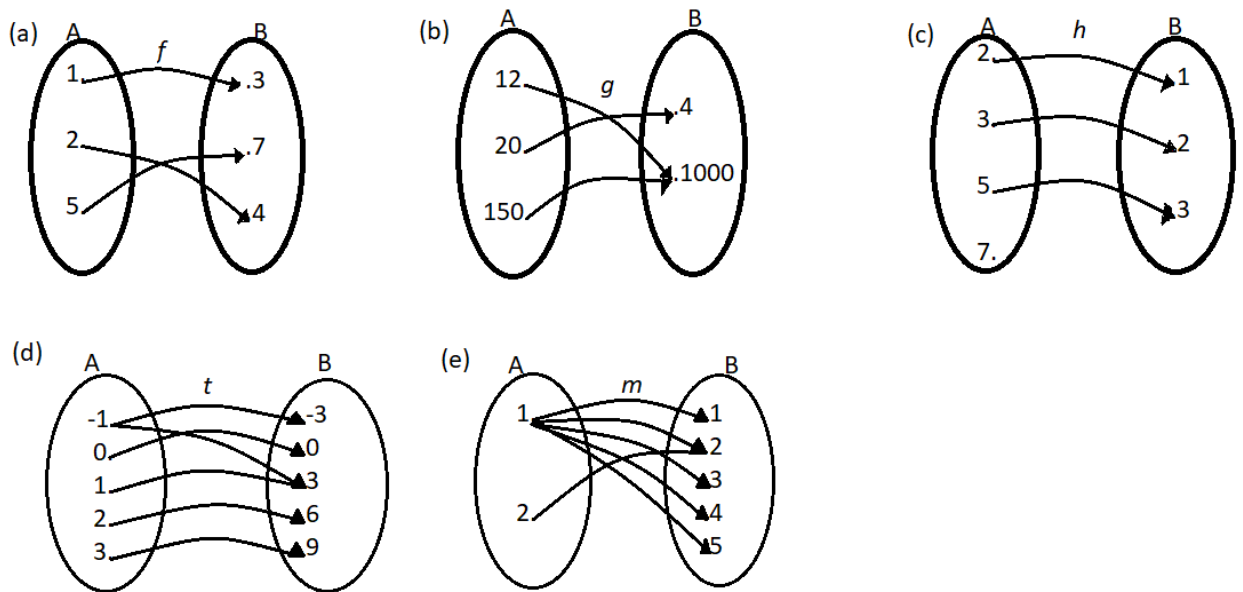


Figura 10 – Fonte: Autor.

As correspondências f e g são funções. Já h , m e t não são funções, pois não relacionam a cada elemento do domínio A um elemento único do contradomínio B .

Definição 3.12. Considere a função $f : A \rightarrow B$. O conjunto imagem da função f é o conjunto de todos os $y \in B$, tais que $y = f(x)$ para algum $x \in A$. Representamos o conjunto imagem da função f por

$$f(A) = \{y \in B; y = f(x), x \in A\}$$

Os casos mostrados no exemplo 12 apresentam *funções reais de variáveis reais*, a saber, do tipo $f : A \subset \mathbb{R} \rightarrow \mathbb{R}$, ou seja, funções com imagens reais cujos domínios são subconjuntos de \mathbb{R} . Neste caso, quando o domínio não é informado, considera-se o maior subconjunto de \mathbb{R} para o qual a função pode ser definida.

Exemplo 12. O domínio da função definida por $f(x) = \frac{1}{x}$ é $D = \mathbb{R} - \{0\}$ e o da função definida por $g(x) = \sqrt{x+2}$ é

$$D = \{x \in \mathbb{R}; x \geq -2\}$$

Proporcionalidade

Definição 3.13. *Dadas duas grandezas, cujas medidas são x e y , dizemos que estas grandezas são diretamente proporcionais se ao multiplicarmos uma delas por um número real, a outra fica também multiplicada por esse número. Quando são inversamente proporcionais, ao multiplicarmos uma delas por um número, a outra fica dividida por esse número.*

Em termos funcionais, uma proporcionalidade é uma função $f : \mathbb{R} \rightarrow \mathbb{R}$ tal que, para quaisquer números reais c , x tem-se $f(cx) = c \cdot f(x)$ (proporcionalidade direta) ou $f(cx) = \frac{f(x)}{c}$, se $c \neq 0$ (proporcionalidade inversa).

Função linear

Definição 3.14. *Uma função $f : \mathbb{R} \rightarrow \mathbb{R}$ é dita linear se existe $a \in \mathbb{R}$ tal que $f(x) = ax$ para todo $x \in \mathbb{R}$.*

A função linear é um modelo matemático para a relação de proporcionalidade, em que $a \in \mathbb{R}$ é dito coeficiente de proporcionalidade.

Numa relação de proporcionalidade, é possível obter o coeficiente a e escrever a função linear que a representa. Para isto, basta tomar valores não nulos das duas medidas e dividir o valor dependente pelo valor independente.

Exemplo 13. *Uma porta analógica associa tensões elétricas de 0 a 5V a sinais analógicos pertencentes ao conjunto $\{0, 1, 2, 3, 4, 5, \dots, 1023\}$ de maneira diretamente proporcional, de modo que o sinal 0 corresponde a 0V e o sinal 1023 corresponde a 5V. Escreva a lei de correspondência da função linear que modela esta proporcionalidade.*

Solução: Considere a função s que dá o sinal analógico em função da tensão elétrica x recebida pela porta. Como $s(5)=1023$, dividindo os valores, temos o coeficiente de proporcionalidade $a = \frac{5}{1023}$. Com isto, a função linear desejada tem lei

$$s(x) = \frac{5}{1023}x.$$

Funções discretas

No trabalho a ser realizado, lidaremos com funções discretas. Quando uma tensão é recebida por uma porta da placa Arduino, a mesma a interpreta como um sinal em forma de número natural. Assim sendo, todos os valores recebidos são discretizados pela placa. Da mesma forma, todos os valores de saída programados são, também

transformados em me de conjuntos discretos, de modo que todas as funções trabalhadas no Arduino Uno são funções discretas, como será definido a seguir.

Definição 3.15. *Uma função real de variável real é discreta se seu domínio e sua imagem são conjuntos discretos.*

Exemplo 14. *A função $f : \mathbb{N} \rightarrow \mathbb{R}$ em que $f(n) = 2n + 1$ é discreta. Observe que \mathbb{N} é discreto e $f(\mathbb{N}) \subset \mathbb{N}$. Logo não contém ponto de acumulação, isto é, $f(\mathbb{N})$ também é discreto. Portanto, f é uma função discreta.*

Exemplo 15. *A função $f : \mathbb{Z} \rightarrow \mathbb{R}$, em que*

$$f(n) = \begin{cases} 0, & \text{se } n = 0 \\ \frac{1}{n}, & \text{se } n \neq 0 \end{cases}$$

não é uma função discreta, pois, pelo que constatamos no exemplo 10, 0 é ponto de acumulação de sua imagem que, portanto, não é discreta.

3.3 Conversão de bases numéricas

O sistema de numeração usual é o sistema posicional decimal, ou seja, de base 10. Assim, a cada ordem posicional corresponde a uma potência de base 10 cujo expoente decresce da esquerda para a direita e cada dígito ou algarismo de um número é o coeficiente que multiplica a potência de base 10 correspondente à ordem que ele ocupa.

Exemplo 16.

$$2895 = 2 \cdot 10^3 + 8 \cdot 10^2 + 9 \cdot 10^1 + 5 \cdot 10^0$$

$$84655 = 8 \cdot 10^4 + 4 \cdot 10^3 + 6 \cdot 10^2 + 5 \cdot 10^1 + 5 \cdot 10^0$$

Números binários

Na computação e na robótica, as informações são recebidas e transmitidas entre os dispositivos em forma de números binários.

Definição 3.16. *Um número binário é um número escrito em base 2, isto é, apenas com os dígitos 0 ou 1.*

É mais fácil compreender como são os números binários quando são convertidos em números de base 10 comuns.

Exemplo 17. *Nas igualdades seguintes, temos, nos primeiros membros, números em base 10 e, no segundo membro, sua escrita em base 2.*

$$0=0, 1=1, 2=10, 3=11, 4=100, 5=101, 6=110, 7=111...$$

Para converter um número de base 10 para um número binário, utilizamos o método das divisões sucessivas, como descrito pelo Algoritmo de Euclides, como é mostrado a seguir:

Seja d o número em base 10 que se deseja converter para binário.

Passo 1: divida d por 2, obtendo um quociente q_1 e um resto r_1 .

$$\begin{array}{r|l} d & 2 \\ r_1 & q_1 \end{array}$$

Passo 2: Caso $q_1 \leq 2$, o número binário desejado é o número formado pelos dígitos $q_1 r_1$, nesta ordem. Caso $q_1 > 2$, passe para o passo 3.

Passo 3: Caso $q_1 > 2$, divida ele por dois, obtendo um quociente q_2 e um resto r_2 . Repita esse processo até que encontre um quociente menor que 2. Seja q_n o primeiro quociente menor que dois encontrado. O número binário desejado é, nesta ordem, $q_n r_n r_{n-1} r_{n-2} \dots r_2 r_1$.

$$\begin{array}{r|l} q_1 & 2 \\ r_2 & q_2 \end{array}$$

Exemplo 18. *Vamos converter o número 37 para sua forma binária utilizando o método das divisões sucessivas, como mostrado no algoritmo.*

37:2 dá 18 e deixa resto 1

18:2 dá 9 e deixa resto 0

9:2 dá 4 e deixa resto 1

4:2 dá 2 e deixa resto 0

2:2 dá 1 e deixa resto 0

Tomando o quociente da última divisão, seu resto e os restos anteriores de maneira recursiva, temos que 37 em binários é 100101.

4 O Arduino e a programação

4.1 Algoritmos de programação

Algoritmos

Um algoritmo é uma sequência finita de passos que, quando seguidos corretamente, findam na conclusão de um objetivo.

Quando se cria um algoritmo, se estabelece um padrão ou padrões de ação para a execução de uma tarefa. Por exemplo, o Algoritmo 1 define a sequência de passos para se trocar um pneu.

Algoritmo 1 Troca de pneu do carro.

- 1: desligar o carro
 - 2: pegar as ferramentas (chave e macaco)
 - 3: pegar o estepe
 - 4: suspender o carro com o macaco
 - 5: desenroscar os 4 parafusos do pneu furado
 - 6: colocar o estepe
 - 7: enroscar os 4 parafusos
 - 8: baixar o carro com o macaco
 - 9: guardar as ferramentas
-

Figura 11 – Fonte: site Christian Cechinel, 2025.

Algoritmo de programação

O algoritmo de programação aqui tratado se referirá a um código que serve para programar um dispositivo ou programa utilizando um software designado para programação. Geralmente, um algoritmo tem três partes:

1. Entrada de dados;
2. Processamento de dados;
3. Saída de dados.

Na primeira parte, que é a entrada, são declaradas as informações necessárias para que o objetivo possa ser alcançado, fazendo com que o dispositivo ou software reconheça essas informações caso elas cheguem a ele. Esse reconhecimento pode ser feito por sensores. Alguns exemplos disto constam no capítulo 5 desta dissertação. Por exemplo, um dispositivo de ventilação reconhece uma determinada temperatura do sistema a ser ventilado através de um sensor de temperatura.

Na parte do processamento de informações, essas informações são transformadas, por meio de expressões algébricas, nas representações adaptadas à linguagem do sistema. Os sensores, por exemplo, recebem informações apenas em forma de tensão elétrica, as quais, os mesmos convertem em um número, o qual eles enviam ao sistema. Através de equações algébricas, é possível converter esse sinal numérico para uma medida que se deseja obter a partir do sensor. Para exemplificar tal processo, considere o sensor HC SR04: um dispositivo que emite e recebe pulsos ultrassônicos e, além disso, consegue cronometrar o tempo entre a emissão e o momento que esse pulso retorna ao dispositivo quando reflete em uma superfície. O sensor fornece ao sistema um número correspondente ao tempo de ida e volta do pulso. Assim, deve-se escrever uma equação que converta este tempo para a distância correspondente em uma unidade de medida conveniente se o objetivo for usar o sensor para medir distâncias.

Na parte da saída de dados, o sistema é programado para se comportar de uma certa maneira a partir da informação obtida e de seu processamento. É nesta fase que a automação se torna realidade, pois o sistema é programado para responder de uma determinada maneira quando recebe um certo estímulo. Usando, ainda, o exemplo do sensor ultrassônico, é possível programar um comando para ativar uma saída de som, luz ou vibração assim que o sensor detectar um objeto a uma determinada distância.

Competências e habilidades elencadas na BNCC

A competência 2 da BNCC diz respeito ao pensamento científico, crítico e criativo. Esta competência traz a visão de que o(a) estudante deve ser preparado para ter a capacidade de resolver problemas com criatividade e investigação. Assim, ele deve desenvolver curiosidade intelectual, além da capacidade de investigar, refletir e resolver problemas de forma lógica e criativa. [5](#)

A habilidade EM13MAT406, da BNCC, propõe que o(a) estudante deve *"utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática."* (BRASIL, 2018)

Com tais propostas em mente, os algoritmos de programação se introduzem no currículo como uma ponte entre a escola e o crescente mundo da tecnologia, no qual os estudantes serão inseridos mais cedo ou mais tarde.

Os algoritmos de programação configuram uma ferramenta para alterar ou criar o funcionamento de algum software ou dispositivo através de uma linguagem de programação.

A BNCC orienta que se utilize conceitos básicos de uma linguagem de programação. Logo, a proposta é que seja algo introdutório e, desta forma, é útil ao professor gerar um interesse nos(as) estudantes. A robótica pode ser um ótimo recurso para tornar a programação algo mais atrativo para o público jovem, por trabalhar também a parte

da montagem dos dispositivos e criar a possibilidade do jovem montar e desenvolver modelos próprios durante as atividades.

4.2 O Arduino

Na cidade de Ivrea, na Itália, o professor Massimo Banzi do Instituto de Design de Interação Ivrea (IDII) teve uma ideia com intuito de viabilizar as criações dos(as) estudante do curso de design: Criar uma placa programável de microcontrolador de baixo custo. Juntamente com ele, Gianluca Martino, David Cuartielles, David Melles e Tom Igoe são os cinco pioneiros do Arduino, cada um em sua área específica do desenvolvimento do sistema. Seu objetivo era criar algo que facilitasse a criação de protótipos e dispositivos eletrônicos e tivesse um preço acessível para estudantes de design e outros cursos na área de tecnologia [10].

O professor Massimo e seus colegas costumavam frequentar uma *pasticceria* local chamada Arduino. O nome foi dado em homenagem ao rei Arduino, que foi rei da Itália de 1002 a 1014 d.C. Antes de se tornar rei, Arduino foi Marquês de Ivrea [14].

Hoje em dia se pode obter um kit básico Arduino por cerca de 180 reais ou simplesmente fazer o seu próprio, pois todos os esquemas de construção de hardware e software estão sob licença pública.



Figura 12 – Fonte: Reddit, 2025

Para fomentar ainda mais o uso da plataforma, o Arduino foi designado para ser uma plataforma *open source*, ou seja, qualquer um pode reproduzir as peças e vender. Assim, é possível adquirir placas nos modelos Arduino de outros fabricantes, o que as torna ainda mais acessíveis [15].

O Arduino é uma plataforma de prototipagem de dispositivos que se baseia na ideia de entrada, processamento e saída, assim como foi tratado no capítulo anterior.

4.2.1 Composição

Podemos dividir o Arduino em duas partes: O hardware (parte física) e o software (parte digital). No hardware, temos as placas Arduino e de conexões protoboard, os fios jumper, as baterias, os diversos tipos de sensores, os LEDs, os Buzzers (dispositivos de saída de som), motores e etc. No software, temos o IDE Arduino, que é um programa utilizado para programar os protótipos criados com o Arduino. Na programação, geralmente, as entradas podem estar em forma de informações advindas de sensores e interpretadas pelo Arduino por meio da programação feita. As saídas são comportamentos do sistema preestabelecidos na programação, que podem ser na forma de um comando executado por um dispositivo de saída como um led ou mais, buzzers, que são dispositivos que emitem som, ou motores.

4.2.2 As portas

As portas do Arduino são conectores enumerados, os quais se diferenciam pela quantidade de níveis que podem interpretar e emitir um sinal.

Na programação, são definidas algumas variáveis, as quais são nomeadas e associadas a alguma porta da placa. Estas variáveis podem ser rotuladas em duas categorias distintas, que são:

- 1- INPUT = entrada de dados;
- 2- OUTPUT = saída de dados.

As portas analógicas só podem ser utilizadas como entradas.

Os sensores, em sua maioria, servem para obter sinais ou informações do ambiente ou dadas pelo utilizador. Assim, quando se conecta um sensor a uma porta, define-se essa porta na programação como *input*, pois ela servirá de entrada de dados. Baseando-se na informação obtida, escolhe-se uma resposta para ser executada pelo dispositivo. Por exemplo, se for uma resposta sonora, utiliza-se um buzzer ou uma caixa de som com alguma mídia de áudio. Se for uma resposta mecânica, conecta-se um motor, se for uma resposta através de luz, conecta-se um LED ou uma cadeia de LED's. Para cada dispositivo de saída, a forma da variável a ser declarada é *output*.

Alguns sensores podem exercer tanto função de entrada quanto de saída, como é o caso do HC SR04, um sensor de pulsos ultrassônicos. Isto por que ele possui dois pinos, nomeados na sequência didática do Capítulo 5 como *trigPin* e *echoPin*. Ele atua como saída quando emite um pulso e como entrada quando capta o retorno do pulso. Cada um desses modos é programado no código especificando o pino do sensor e a porta à qual o mesmo está conectado.

Os dados que são transmitidos pelas portas do Arduino são nada mais que uma tensão elétrica modulada que vai de 0V a 5V. As portas interpretam a tensão recebida

como um número (byte) que pode variar dentro de um conjunto de números naturais ou zero (bytes), a depender da quantidade de bytes que podem ser processados pela porta.

O Arduino UNO tem três diferentes tipos de portas, os quais se diferenciam pela quantidade de bytes diferentes que esta porta consegue processar. Tal quantidade é sempre uma potência de base 2, pois os códigos interpretados são números binários e, assim sendo, se um binário tem n dígitos, ele pode ser feito de 2^n modos. O maior e o menor número natural nesta representação são 0 e $2^n - 1$, respectivamente. Se $n = 3$, por exemplo, eles serão 0 e $2^3 - 1 = 7$. Cada dígito é chamado de bit.

Denominamos de porta de n bits a uma porta que interpreta códigos binários de n dígitos.



Figura 13 – Fonte: eBay, 2025.

No Arduino UNO, mostrado na figura 13, que será utilizado para a sequência didática, existem 3 tipos diferentes de porta, que são:

- Portas digitais (2 bytes de 1 bit);
- Portas PWM (256 bytes de 8 bits)
- Portas analógicas. (1024 bytes de 10 bits)

As portas digitais processam bytes de 1 bit. Estes bytes podem ser 0 ou 1. Na prática, estas portas são utilizadas na função de ligar ou desligar um dispositivo. Um

botão de liga desliga qualquer é uma entrada de uma porta digital. No Arduino UNO, as portas digitais são as portas 0,1,2,4,7,8,12 e 13, sendo que 0 e 1 são portas que devem ser evitadas, pois são mais destinadas para comunicação serial.

As portas PWM, processam bytes de 8 bits. Um byte de 8 bits pode ser de $2^8 = 256$ modos distintos. Assim sendo, uma porta PWM pode interpretar 256 níveis de tensão elétrica e associar a bytes que vão de 00000000 a 11111111, que correspondem, respectivamente a 0 e 255. Esta associação, feita pela porta PWM é crescente. Isto significa que o sinal 0 corresponde à tensão 0V, enquanto que o sinal 255 corresponde à tensão 5V. Assim, elas podem modular a tensão de 5V em $2^8 = 256$ níveis graduais. Nesta porta, pode-se alterar a tensão de uma saída deixando-a mais intensa ou menos intensa. Por exemplo, um LED pode oscilar seu brilho do mais forte ao mais fraco através de todos esses níveis. No Arduino UNO, as portas PWM têm um sinal de ~ antes da numeração, que são as portas ~3, ~5, ~6, ~10 e ~11.

As portas analógicas interpretam 1024 bytes distintos de 10 bits cada. Então, a tensão elétrica que chega a ela pode ser interpretada em 1024 níveis distintos, que vão de 000000000=0 a 111111111=1023, os quais correspondem, respectivamente às tensões de 0V e 5V. Nesta porta, os incrementos na tensão podem ser bem mais minuciosos, além de ter uma sensibilidade maior a variações de tensão.

A linguagem de programação do Arduino

Na programação de um dispositivo com Arduino, utiliza-se uma linguagem variante da linguagem de programação C++, com algumas adaptações e com bibliotecas próprias que possibilitam a interação com a placa e seus componentes.

O software de programação do Arduino chama-se Arduino IDE, cuja sigla significa *Integrated Development Environment*, que quer dizer Ambiente de desenvolvimento integrado. Neste software, temos diversas opções para alterar e adaptar dispositivos com placas Arduino. Além disso, a cada nova placa Arduino que é lançada, o IDE Arduino é atualizado e disponibilizado gratuitamente no site oficial.

Todo código para Arduino se estrutura em três partes, colocadas na ordem a seguir:

- Preâmbulo: Declaração de variáveis, constantes, portas e inclusão de bibliotecas;
- *void setup* (Configuração de portas): Ambiente de configuração de portas, onde cada porta declarada no preâmbulo tem sua função determinada;
- *void loop* (Configuração do comando cíclico): Configuração do algoritmo a ser executado pelo dispositivo, cuja sequência permanece se repetindo enquanto o dispositivo tiver energia.

Na figura 14 abaixo, segue um exemplo de código para um dispositivo no Arduino IDE.

```
1  int LED = 13; //Declaração de uma variável chamada LED conectada à porta 13
2  void setup() { //Função de ajuste executada uma única vez
3  pinMode (LED, OUTPUT); //Rotula a variável como saída de dados
4  }
5  void loop() {
6  digitalWrite(LED, HIGH);
7  //Torna a variável LED ligada (HIGH coloca a tensão no máximo: 5 V)
8  delay(1000);
9  //Espera um segundo (1000 milissegundos)
10 digitalWrite(LED, LOW);
11 // Torna a variável LED desligada (LOW coloca a tensão no mínimo: 0 V)
12 delay(1000); }
```

Figura 14 – Fonte: Autor. Feito com auxílio do software Arduino IDE.

No exemplo acima, o preâmbulo corresponde apenas à primeira linha, onde é declarada a variável LED conectada à porta 13. Isto, porque na montagem, um led foi conectado à porta digital 13 da placa Arduino. O *void setup* é feito nas linhas 2, 3 e 4, onde a variável LED é configurada como saída de dados, pois o LED funciona como saída, já que o sistema dá o comando para que o mesmo emita luz. O *void loop* está nas demais linhas, que vão da linha 5 à linha 12. Nelas, o algoritmo de programação que está escrito, em língua comum, significa:

- 1-Acender o LED na tensão mais alta, (5V);
- 2-Esperar 1000 milissegundos e ir para o passo 3;
- 3-Apagar o LED (0V);
- 4-Esperar 1000 milissegundos.

Como este algoritmo está escrito no ambiente *void loop*, após o quarto passo, ele retorna ao primeiro e, assim, sucessivamente, criando um comportamento cíclico.

O monitor serial

Assim como o Arduino pode ser utilizado para programar sensores, ele também pode apresentar os valores correspondentes aos sinais recebidos pelos sensores programados. O monitor serial é uma aba que pode ser executada dentro do Arduino IDE, e nela, pode-se exibir valores a partir de sinais recebidos pelos sensores que estiverem programados e acoplados à placa Arduino selecionada no Arduino IDE. O modo pelo qual se faz isso se dá da seguinte maneira: o sinal recebido pelo sensor chega na porta selecionada na forma de uma tensão elétrica e é interpretado pela porta como um byte de 1, 8 ou 10 dígitos (como tratado anteriormente), que é automaticamente convertido

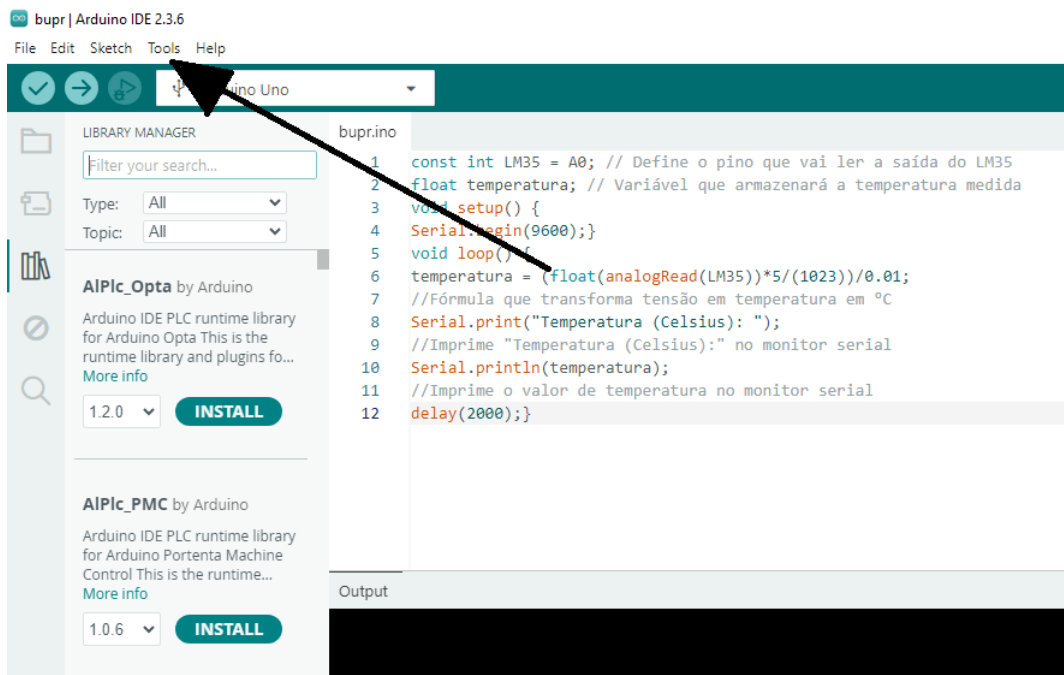


Figura 15 – Fonte: Autor. Feito com auxílio do software Paint.

pelo Arduino em seu número correspondente de base 10. Este número pode ser apresentado no monitor serial diretamente ou ser transformado na medida da grandeza que o sensor capta juntamente com a unidade de medida desejada. Esta transformação consiste numa função matemática que é escrita no código representando a equação com os respectivos caracteres adaptados à linguagem de programação do Arduino.

Para abrir o monitor serial, deve-se seguir os seguintes passos:

- 1- Abra o Arduino IDE;
- 2- Clique em Tools, ou Ferramentas;
- 3- Clique em Serial monitor, ou monitor serial.
- 4- O monitor serial será aberto e, por fim, você verá as informações nele impressas.

O multímetro digital

Juntamente ao kit Arduino, é importante que se disponha de um multímetro digital. Este aparelho servirá para averiguar as condições dos componentes do kit, assim como verificar se existe mal contato em algum conector, além de verificar condições das baterias e resistores.

O multímetro é um instrumento de medida utilizado para medir tensão, corrente e resistência elétricas. Além disso, o multímetro pode ser utilizado para verificar conexão elétrica entre dois pontos.

O multímetro digital contém o aparelho e dois cabos, sendo eles, um de cor preta e outro de cor vermelha denominados, respectivamente, de ponta de prova preta e ponta

de prova vermelha. Além disso, o multímetro possui três entradas nas quais os cabos são conectados a depender do tipo de medição a ser realizada, sendo elas:

1-V Ω mA : Utilizada para medições de tensão elétrica (V), resistência elétrica (Ω) e corrente elétrica, (A).

2-10A máx.: Porta designada para medir correntes elétricas de amperagem mais elevada, as quais podem ser de até 10 Ampères.

3-COM: Destinada para a ponta de prova preta. Serve para medir a referência negativa ou terra do circuito.



Figura 16 – Fonte: Site da Amazon.

Medição de tensão elétrica

Para medir tensão elétrica, deve-se plugar o cabo vermelho na porta V Ω mA e apontar o cursor para o menor limite máximo de tensão a ser medida, que no multímetro, vai de 200 mV a 500 V.

Medição de corrente elétrica

Para medir corrente elétrica, deve-se plugar o cabo vermelho na porta V Ω mA. O cursor deve estar ajustado à menor corrente possível, sendo que deve-se apontá-lo para 20 mA para medir correntes de até 20 mA, 200 mA para correntes de até 200 mA, e assim por diante.

5 Sequência didática

5.1 Aplicação

Neste capítulo, propomos uma sequência didática que trata de funções e algoritmos de programação utilizando, como ferramenta, a plataforma de prototipagem Arduino. As atividades envolverão, desde o início, habilidades matemáticas trabalhadas em sala de aula, entre as quais, nas atividades propostas, destacaremos a modelagem matemática utilizando funções.

Alguns pré-requisitos para a realização desta sequência são que a escola disponha de um espaço com, pelo menos, 8 computadores em bom funcionamento, além de disponibilizar para os(as) estudantes, 8 kits básicos Arduino Uno. Desta forma, o professor pode remanejar a turma para o espaço ou laboratório de informática, ou ainda, trazer os computadores e os kits para a sala de aula. Além disso, ao aplicar esta sequência para mais de uma turma, é possível notar que cada turma irá, provavelmente, progredir em ritmos diferentes. Assim, cabe ao professor investigar maneiras diferentes de planejar as aulas a depender da turma a que se destina a aula.

As programações e montagens propostas nesta sequência foram feitas a partir da apostila *Robótica livre com Arduino*, desenvolvida pela Secretaria de Educação do Estado de Pernambuco, com o auxílio de uma equipe de técnicos em programação e mecatrônica Arduino. [9](#)

Sendo possível a aplicação desta sequência, os(as) estudantes vão desenvolver a habilidade de modelar padrões matematicamente usando funções discretas, aprender a utilizar uma linguagem de programação para implementar algoritmos que envolvem tais funções, com o intuito de por em prática um objetivo, além de adquirir experiência na manipulação e programação de diversos sensores e saídas do Arduino. Isto possibilitará o desenvolvimento de projetos que podem ser executados pelos(as) estudantes com a assistência do professor. Tais projetos podem ser levados a mostras e exposições de robótica para serem apresentados ao público. A participação dos(das) estudantes em exposições de robótica e tecnologia é uma experiência enriquecedora, pois nesses eventos, os(as) estudantes têm contato com pessoas que atuam na área de tecnologia, além de ficar a par dos trabalhos e ideias de outras equipes de estudantes, o que pode incentivá-los fortemente a continuar com o estudo e as criações.

Vejam, a princípio, quais habilidades da BNCC tratam de programação e de funções. Para o Ensino Médio, a BNCC propõe as seguintes habilidades [5](#):

(EM13MAT406) Utilizar os conceitos básicos de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.

Figura 17 – Fonte: BNCC, p. 531,(2025).

(EM13MAT501) Investigar relações entre números expressos em tabelas para representá-los no plano cartesiano, identificando padrões e criando conjecturas para generalizar e expressar algebricamente essa generalização, reconhecendo quando essa representação é de função polinomial de 1º grau.

Figura 18 – Fonte: BNCC, p. 533,(2025).

Como vemos na habilidade da figura 16, a BNCC propõe que o(a) estudante aprenda a utilizar os conceitos básicos de uma linguagem de programação para implementar algoritmos. Nesta sequência didática, os algoritmos a serem executados pelos dispositivos são escritos na linguagem de programação do Arduino, sendo, desta forma implementados aos dispositivos.

Na habilidade *EM13MAT501*, é proposto que o(a) estudante investigue relações entre números para expressar esta relação algebricamente, reconhecendo quando é uma função polinomial de 1º grau. Esta habilidade é contemplada nos problemas propostos ao(a) estudante, que incluem escrever as funções discretas* ¹ que transformam as tensões recebidas pelas portas em números naturais.

(EM13MAT302) Resolver e elaborar problemas cujos modelos são as funções polinomiais de 1º e 2º graus, em contextos diversos, incluindo ou não tecnologias digitais.

Figura 19 – Fonte: autor, a partir da BNCC, p. 534,(2025).

A maioria dos protótipos a serem programados nesta sequência didática têm como modelo matemático para seu funcionamento uma função. A programação dos protótipos robóticos traz uma diversificação no contexto rotineiro de sala de aula, levando o(a) estudante a conhecer tarefas diferentes, relacionadas à tecnologia e que envolvem diretamente seus conhecimentos matemáticos. Desta forma, o(a) estudante reflete sobre a relevância daquilo que estuda dia após dia. A inclusão das tecnologias digitais no ensino é frequentemente endossada pelo currículo, como vemos no texto da BNCC.

¹ As funções discretas consideradas têm leis de correspondência dadas por fórmulas do primeiro grau.

A competência específica 4 da BNCC propõe que o(a) estudante deve aprender a:

Compreender e utilizar, com flexibilidade e fluidez, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas, de modo a favorecer a construção e o desenvolvimento do raciocínio matemático.

(BNCC, 2018)

Logo em seguida, tratando desta competência, a BNCC propõe que

As habilidades vinculadas a essa competência tratam da utilização das diferentes representações de um mesmo objeto matemático, tendo em vista que elas têm um papel decisivo na aprendizagem dos(das) estudantes. Ao conseguirem utilizar as representações matemáticas, compreender as ideias que elas expressam e, quando possível, fazer a conversão entre elas, os(as) estudantes passam a dominar um conjunto de ferramentas que potencializa de forma significativa a capacidade de resolver problemas, comunicar e argumentar; enfim, ampliar a capacidade de pensar matematicamente. Além disso, a análise das representações utilizadas pelos(as) estudantes para resolver um problema permite compreender os modos como o interpretaram e como raciocinaram para resolvê-lo.

(BNCC, 2018)

Nas atividades com Arduino, o(a) estudante é constantemente exposto a desafios onde ele precisa representar regras matemáticas em formas distintas, além de ter que utilizar conhecimentos e técnicas matemáticas, como a manipulação de funções para resolver os problemas e programar corretamente os protótipos, que vai ao encontro do que é proposto nessa competência.

A programação de protótipos de automações, como é ensinada na sequência, tem como um de seus objetivos a resolução de problemas reais, ou seja, o desenvolvimento de facilidades que melhorem a qualidade de vida das pessoas. A utilização da matemática para este fim é uma forma de contemplar o que o currículo propõe, de utilizar as representações matemáticas para potencializar a capacidade de resolver problemas. Neste sentido, além da competência 4, temos também a competência específica 3, que

diz o seguinte:

Utilizar estratégias, conceitos e procedimentos matemáticos, em seus campos – Aritmética, Álgebra, Grandezas e Medidas, Geometria, Probabilidade e Estatística –, para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente.

(BNCC, 2018)

Como as linguagens de programação são um tema novo no Ensino Médio, a quantidade de trabalhos nesse sentido é limitada. Sendo assim, propomos uma sequência atrativa para o(a) estudante, por incluir robótica, além de contemplar aquilo que é perdido pelo currículo: uma forma de resolver problemas em diversos contextos utilizando modelos matemáticos. Esta sequência de atividades enfatiza o emprego de funções discretas na programação de sensores e saídas com a plataforma Arduino, mesclando assim as habilidades anteriormente citadas.

O público alvo deste projeto são os(as) estudantes do Ensino Médio de 14 a 18 anos. Para aplicar totalmente esta sequência, seriam necessárias, pelo menos, 24 aulas de 50 minutos. Assim, em escolas regulares, onde não há tanta disponibilidade de tempo, o professor pode escolher uma ou mais atividades dentre as que são de montagem e programação, que forem mais pertinentes para o conteúdo matemático a ser abordado. Além disso, as atividades com o multímetro são dispensáveis.

Dividiremos esta sequência em quatro partes:

- I:** Introdução ao tema;
- II:** Componentes básicos e o multímetro;
- III:** Programação de protótipos;
- IV:** Trabalho de prototipagem em grupo.

5.2 Sequência didática

Descrição

- Título: Funções discretas e algoritmos de programação na robótica com Arduino.
- Professor: Carlos Gustavo Barreto de Farias Júnior.
- Público-alvo: 1º, 2º e 3º anos do Ensino Médio.

- Duração: De 8 a 24 horas-aula.

Habilidades

- Conhecer e empregar os conceitos básicos da linguagem de programação Arduino;
- Manipular funções discretas para obter um resultado determinado;
- Utilizar os modelos matemáticos adaptados à linguagem de programação do Arduino para regular o funcionamento de sensores e saídas;
- Utilizar as funções matemáticas, a montagem e a linguagem de programação Arduino para criar protótipos de facilidades e soluções para problemas reais para melhorar a qualidade de vida das pessoas.

Recursos e materiais

- Computadores;
- Kits básicos Arduino;
- Apostilas, Cadernos e canetas;
- Quadro branco, lápis pilotos e apagador;
- Data show e notebook.
- Multímetros

5.2.1 Primeiro encontro: Introdução ao Arduino e ao multímetro

Na primeira aula, os(as) estudantes devem se organizar em grupos de 4 participantes. Espera-se que esta etapa transcorra em 2 horas aula.

Atividade 1: Distribuir um kit Arduino e um multímetro para cada grupo. A partir daí, apresentar cada componente do kit aos(às) estudantes, utilizando as informações contidas na seção sobre o Arduino. Explicar a importância dos resistores para proteger os circuitos e os componentes oferecendo resistência à corrente elétrica, e etc... Na sequência, o professor deve discorrer brevemente sobre a história da criação do Arduino. (Ver o capítulo 4)

Atividade 2: Propor que os(as) estudantes conectem o cabo vermelho em $V\Omega mA$ e o cabo preto em COM. (Alguns multímetros tem entradas diferentes para Volts e para Ampères. Assim, deve-se conectar na entrada específica do que se quer medir.) Na sequência, orientar aos(às) estudantes que posicionem o cursor do multímetro em 20V. Assim que todos os grupos tiverem configurado o cursor, propor que cada grupo posicione as pontas dos cabos uma em cada polo da bateria disponível no kit Arduino, e registre no caderno o resultado mostrado no display do multímetro. Apresentar o problema 5.1

Problema 5.1. *Uma bateria pode ser classificada como nova, semi-nova ou usada a partir da tensão medida entre seus pólos. Para classificar, utiliza-se o seguinte critério:*

A bateria está

- (i) Nova, se a tensão medida é maior que a informada no rótulo;*
- (ii) Semi-nova, se a tensão medida é igual à informada no rótulo;*
- (iii) Usada se a tensão medida é menor que a informada no rótulo.*

Utilizando este critério, classifique a bateria do seu kit.

Atividade 3: Após a conclusão do problema 5.1, propor aos(às) estudantes o Problema 5.2

Problema 5.2. *Faça a medição da corrente elétrica entre os pólos da sua bateria e anote o valor em seu caderno. (Para medir a corrente elétrica, basta posicionar as pontas do multímetro uma em cada polo da bateria e apontar o cursor para a faixa de 20 miliAmpères (20 mA))*

5.2.2 Segundo encontro: Medição de resistência elétrica

Neste, e em todos os próximos encontros, a turma deve ficar dividida nos mesmos grupos de quatro integrantes. O tempo estimado para este encontro é de duas horas-aula.

Atividade 1: Propor que cada grupo pegue um resistor do kit Arduino e entregar uma folha com o código de cores dos resistores, disponível na figura 19 e o texto explicativo sobre o código de cores. Na sequência, apresentar o texto explicativo e propor o Problema 5.3.

Código de cores dos resistores: Texto explicativo

O código de cores dos resistores consiste numa sequência de faixas coloridas pintadas no resistor que significa o número $d_1d_2 \cdot 10^p$ variando em v percentuais, onde d_1d_2 é um número inteiro de dois algarismos que correspondem à primeira e segunda cor, respectivamente, o qual é multiplicado pela potência p de 10, correspondente à terceira cor. O número resultante é a resistência oferecida pelo resistor, a qual pode variar na porcentagem v correspondente à quarta cor informada no resistor.

O código de cores a seguir mostra o que significa cada cor. Os estudantes devem anotar as cores e, baseando-se no código, escrever a sequência de dígitos e determinar a resistência correspondente do resistor.

Cor	Dígitos Significativos (1 e 2)	Multiplicador (3)	Tolerância (4)
Preto	0	10^0	
Marrom	1	10^1	$\pm 1\%$
Vermelho	2	10^2	$\pm 2\%$
Laranja	3	10^3	
Amarelo	4	10^4	
Verde	5	10^5	
Azul	6	10^6	
Violeta	7	10^7	
Cinza	8	10^8	
Branco	9	10^9	
Ouro	-	10^{-1}	$\pm 5\%$
Prata	-	10^{-2}	$\pm 10\%$
Sem Cor	-	-	$\pm 20\%$

Figura 20 – Fonte: Apostila Robótica Livre com Arduino - p. 14

Exemplo 19. Considere o resistor na figura 20.



Figura 21 – Fonte: Apostila Robótica Livre com Arduino - p. 14

A sequência de cores deste resistor é verde, azul, vermelho e ouro. Com isto, ele tem a resistência

$$56.10^2 = 5600\Omega$$

variando em $\pm 5\%$.

Problema 5.3. *Utilizando as instruções do texto explicativo na folha entregue pelo professor, interprete o código de cores do seu resistor e escreva o significado em seu caderno. Na sequência, aponte o cursor do multímetro para $20k\Omega$ e meça a resistência de seu resistor. Compare o resultado da medição com o significado do código. Qual foi maior? O código está correto?*

Atividade 2 - Propor que os(as) estudantes peguem a placa protoboard no kit e apontem o cursor do multímetro para o marcador que contém um símbolo de uma pequena buzina. Ao apontar o cursor para este ponto, o multímetro consegue detectar se entre as pontas dos cabos existe condução elétrica e confirma isto com um sinal sonoro. Desta forma, é possível verificar se entre dois pontos de uma placa existe ou não uma ligação. Na sequência, propor para os(as) estudantes o Problema 5.4.

Problema 5.4. *Utilize o multímetro no modo da verificação de circuito (buzina) para verificar como os pontos da placa protoboard se conectam. Com isto, faça um desenho em seu caderno ilustrando como são as ligações internas da placa protoboard, seguindo o modelo na figura 21.*

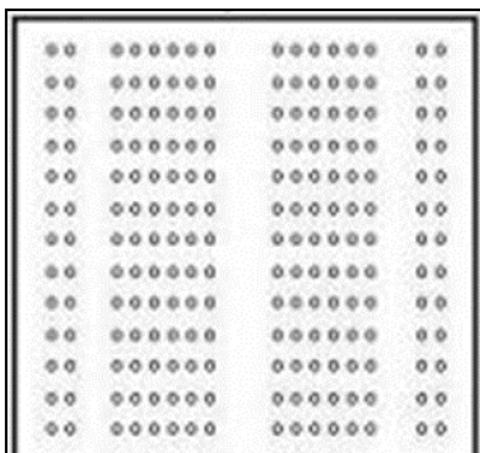


Figura 22 – Fonte: Apostila Robótica livre com Arduino, p. 15

5.2.3 Terceiro encontro: Função da porta digital e o Pisca com LED

Atividade 1: Organizar os grupos nos quais a turma foi dividida desde o início da seguinte maneira: cada integrante deverá ter uma função das

listadas a seguir:

1-Montador: É responsável pela montagem dos protótipos e pelos componentes disponibilizados.

2-Programador: É Responsável pela programação, ou seja, escrita, compilação e carregamento dos códigos para a placa Arduino.

3-Líder: É Responsável por garantir que cada um exerça sua função específica.

4- Almojarifado: Fica responsável por receber e devolver o material que o professor entregar ao grupo.

(Esta divisão deve continuar para todos as atividades até o final desta sequência, porém com revezamento das funções entre os membros do grupo.)

Atividade 2: Propor que os(as) estudantes liguem os computadores, iniciem o Arduino IDE e entregar a cada grupo um material de apoio com as figuras 22 e 23 impressas. Na sequência, propor que montem e programem o dispositivo como nas figuras 22 e 23.

Esquema de Ligações na Protoloard:

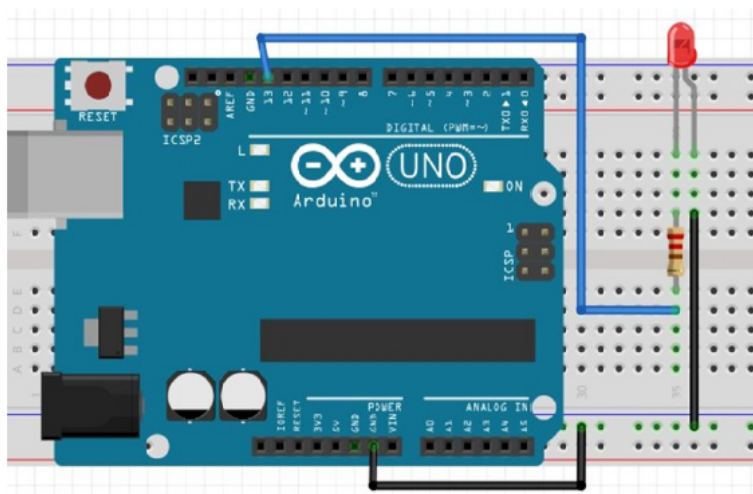


Figura 23 – Fonte: Apostila Robótica livre com Arduino, p. 7.

```

1  int LED = 13; //Declaração de uma variável chamada LED conectada à porta 13
2  void setup() { //Função de ajuste executada uma única vez
3  pinMode (LED, OUTPUT); //Rotula a variável como saída de dados
4  }
5  void loop() {
6  digitalWrite(LED, HIGH);
7  //Torna a variável LED ligada (HIGH coloca a tensão no máximo: 5 V)
8  delay(1000);
9  //Espera um segundo (1000 milisegundos)
10 digitalWrite(LED, LOW);
11 // Torna a variável LED desligada (LOW coloca a tensão no mínimo: 0 V)
12 delay(1000); }

```

Figura 24 – Fonte: Apostila Robótica livre com Arduino, p. 7.

Atividade 3: Explicar aos(às) estudantes o funcionamento da porta digital, onde o LED foi conectado na atividade anterior. Explicar que a porta digital executa apenas dois comandos, que são 1 ou 0, que são interpretados pela porta como sinal alto (5V) ou sinal baixo (0V), respectivamente, e desenhar no quadro o diagrama da figura 24. Na sequência, proponha o problema 5.5.

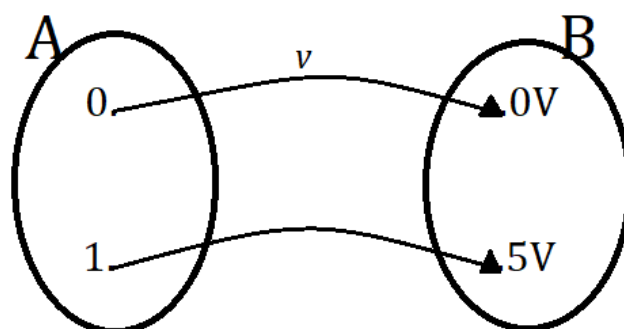


Figura 25 – Fonte: autor

Problema 5.5. Determine uma relação matemática que defina o funcionamento da porta digital, que transforma os sinais 0 e 1 nas tensões 0V e 5V, respectivamente.

Atividade 4: Mostrar aos(às) estudantes a seguinte resolução: Defina-se uma função $v : \{0, 1\} \rightarrow \mathbb{R}$ na qual $v(n) = 5n$, onde o valor $v(n)$, dado em volts (V), é a tensão resultante do valor n atribuído à porta. Após isto, propor o problema 5.6.

Problema 5.6. Faça as seguintes alterações no protótipo do pisca com LED:

- (a) LED com sinal alto por 1 segundo e baixo por 1 segundo;
- (b) LED com sinal alto por 3 segundos e baixo por 1 segundo;
- (c) LED com sinal alto por 1,5 segundo e baixo por 3,2 segundos;
- (d) Utilize outra porta digital para o LED e faça-o funcionar nela.

Atividade 5: Propor o problema 5.7

Problema 5.7. Segundo a biologia, o olho humano consegue perceber o piscar de uma luz se ele ocorrer num tempo maior que ou igual a 40 milissegundos. Faça esse teste usando o LED Arduino.

O professor deve verificar, de grupo em grupo, como os(as) estudantes estão desenvolvendo as atividades e dar o suporte necessário.

Caso as aulas não sejam geminadas, é importante orientar os(as) estudantes a salvar os códigos escritos numa pasta com o nome do grupo.

5.2.4 Quarto encontro: Controlar LED com sinal PWM

Nesta atividade, os(as) estudantes utilizarão portas PWM, que são de 8 bits, para ativar o LED. Esta atividade tem como objetivo mostrar como uma porta PWM pode modular a tensão indo da mais baixa para a mais alta passando por vários níveis.

Função de uma porta PWM

Atividade 1: Explicar aos(às) estudantes o funcionamento básico da modulação do sinal por uma porta PWM, que o sinal modulado pela PWM vai de $00000000 = 0$ a $11111111 = 255$, os quais correspondem a tensões de 0V a 5V, respectivamente. Na sequência, propor o problema 5.8

Problema 5.8. Sabendo que a tensão emitida no LED é proporcional ao sinal obtido pela porta PWM, de maneira que os sinais 0 e 255 (maior e menor sinal) correspondem às tensões 0V e 5V (tensão mínima e tensão máxima), respectivamente, determine a função crescente que transforma o sinal PWM n na voltagem $v(n)$ da porta Arduino. Ilustre esta função num diagrama.

Solução: O domínio da função v é composto pelos valores dos sinais recebidos pela porta PWM, ou seja, se trata do conjunto $\{0, 1, 2, \dots, 255\}$. Como a tensão $v(n)$ é proporcional a n e $v(0) = 0$, a função v é do tipo $v(n) = an$, em que a é a constante de proporcionalidade, a qual pode ser obtida a partir de um sinal $n \neq 0$, pela fórmula

$$a = \frac{v(n)}{n}.$$

Utilizando valores dados na questão, obtemos

$$a = \frac{5}{255} = \frac{1}{51}.$$

Então, a função desejada é $v : \{0, 1, 2, \dots, 255\} \rightarrow \mathbb{R}$, na qual $v(n) = \frac{1}{51} \cdot n$, onde $v(n)$ é a voltagem emitida pela porta para a saída acoplada, em função do valor n atribuído à porta. O diagrama a seguir ilustra tal função.

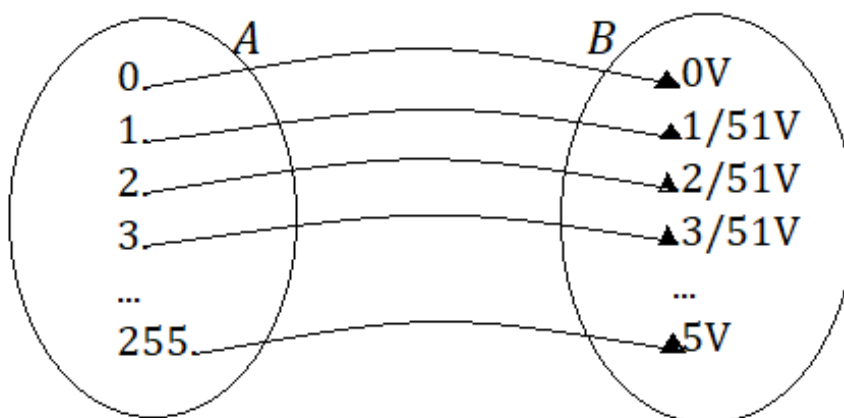


Figura 26 – Fonte: Autor.

Atividade 2: Propor a montagem e programação do dispositivo de Pisca com led da figura 26 com o código da figura 27.

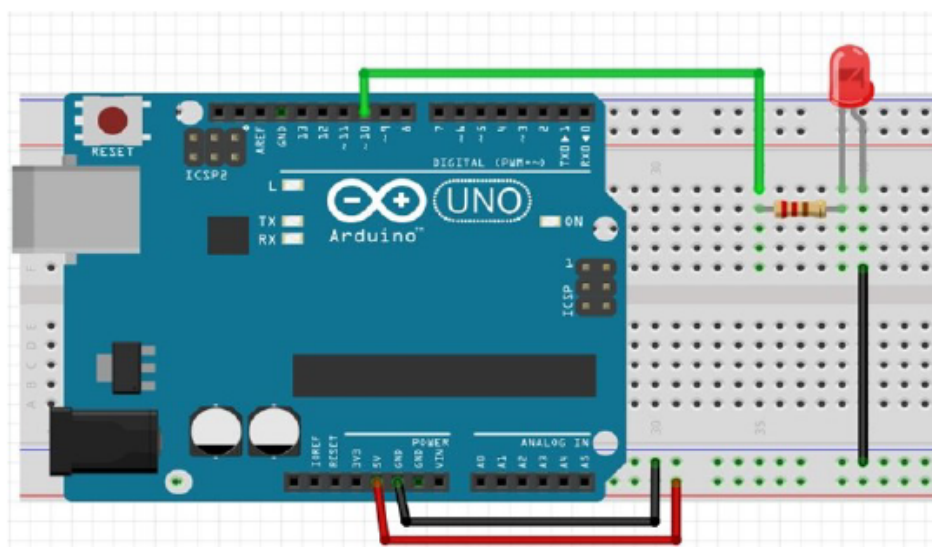


Figura 27 – Fonte: Apostila Robótica livre com Arduino, p.13.

```
1  int led=10; //Declara a variável led na porta 10;
2  int brilho=0; // Fixa o valor inicial da variável brilho = 0;
3  int pulo=5; // Fixa a constante pulo com o valor 5.
4  void setup () {
5  |   | pinMode (led, OUTPUT);} //Configura led como uma saída.
6  void loop () {
7  analogWrite (led, brilho); // Atribui à variável led o valor da variável brilho;
8  brilho = brilho + pulo;
9  // Define que o valor de brilho será dado por seu valor atual mais pulo;
10 if (brilho==0 || brilho==255){pulo= - pulo;}
11 //Se brilho = 0 ou brilho 0 255, o valor de pulo muda de sinal. (||="ou")
12 delay(30);} //Espera 30 milissegundos.
```

Figura 28 – Fonte: Autor.

Atividade 3: Propor os problemas 5.9, 5.10 e 5.11

Problema 5.9. *Altere o código no Arduino IDE para que o led oscile*

- (a) *mais rápido.*
- (b) *mais lento.*

Problema 5.10. *Escreva de maneira abreviada a progressão aritmética finita com os valores atribuídos à variável brilho no primeiro momento do funcionamento, ou seja, quando a tensão está aumentando de 0 a 255.*

Problema 5.11. *Considere a P.A do item anterior. Seja a_n seu termo geral.*

- (a) *Determine a_{10} .*
- (b) *Determine, se possível, a_{25} e a_{40} .*
- (c) *Escreva uma fórmula para encontrar um termo qualquer a_n em função de n .*
- (d) *Quantos termos tem a P.A. em questão?*
- (e) *Calcule a soma de todos os termos desta P.A..*

5.2.5 Quinto encontro: Controlar um pisca de LED com potenciômetro

Nos dois encontros anteriores, controlamos a saída LED utilizando sinal digital e sinal PWM. Nesta atividade, controlaremos a saída LED utilizando sinal analógico. Para controlar o sinal analógico, utilizaremos um potenciômetro, que é um botão giratório que servirá como um controle de intervalo do pisca de LED.

Atividade 1: Explicar aos estudantes o funcionamento básico da modulação do sinal por uma porta analógica, que vai de $0000000000 = 0$ a

1111111111 = 1023, os quais correspondem a tensões de 0V a 5V, respectivamente. Na sequência, propor o problema 5.12

Problema 5.12. *Se os sinais 0 e 1023 da porta analógica correspondem a 0V e a 5V, respectivamente, determine a função que transforma o sinal analógico n na voltagem $v(n)$ da porta Arduino. Ilustre esta função num diagrama.*

Solução: É a função $v : A \rightarrow \mathbb{R}$, na qual $A = \{0, 1, 2, \dots, 1023\}$ e $v(n) = \frac{5}{1023}n$, onde $v(n)$ é a voltagem emitida pela porta para a saída acoplada, em função do valor n atribuído à porta.

Atividade 2: Entregar a cada grupo os componentes listados a seguir e propor a montagem e programação do protótipo nas figuras 28 e 29.

Componentes

- 1 led;
- 1 resistor de 1000 Ohms;
- 3 jumpers pretos;
- 4 jumpers coloridos de cores distintas;
- 1 placa protoboard;
- 1 potenciômetro;
- 1 placa Arduino Uno.

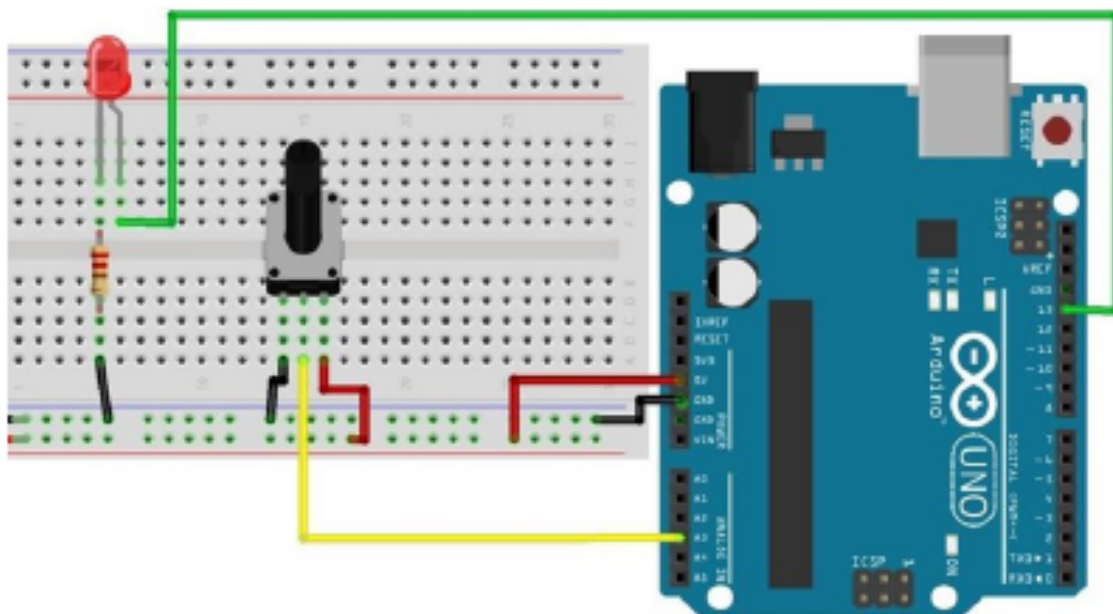


Figura 29 – Fonte: Apostila Robótica livre com Arduino, p. 20

```
1  int sensorPin = A3; // Seleciona o pino de entrada do potenciômetro;
2  int ledPin = 13; // Seleciona o pino do LED ;
3  int sensorValue =0;
4  //Configura a variável sensorValue, que carrega o valor do potenciômetro.
5  void setup() {
6  Serial.begin(9600);
7  // Inicia comunicação com o monitor serial;
8  pinMode(ledPin, OUTPUT); }
9  // Declara ledPin como uma saída}
10 void loop() {
11 sensorValue = analogRead(sensorPin);
12 //Lê o valor do sensor e o atribui a sensorValue;
13 Serial.println(sensorValue); // Imprime <sensorValue> na serial
14 digitalWrite(ledPin, HIGH); // Liga o LED
15 delay(sensorValue); // mantém o comando por <sensorValue> milissegundos
16 digitalWrite(ledPin, LOW); // Desliga o LED
17 delay(sensorValue); // Mantém o comando por <sensorValue>
18 }
```

Figura 30 – Fonte: Autor. Feito com auxílio do software Arduino IDE.

A porta A3 é a porta analógica escolhida para ser a porta de entrada onde será conectado o potenciômetro, declarado no código como `sensorPin` (Pino de sensor). O Led, assim como foi feito no terceiro encontro, é declarado numa porta digital, pois vai apenas ligar ou desligar e, a última declaração é da variável `sensorValue`, que assumirá o valor dado pelo Potenciômetro, de 0 a 1023. O código novamente declara `ledPin` como uma saída em `pinMode(ledPin, OUTPUT)`. Em “`sensorValue = analogRead(sensorPin)`”, é dado o comando de ler o valor do sensor potenciômetro, que será representado por “`sensorValue`”. Na linha seguinte, “`Serialprintln(sensorValue)`”, o valor do potenciômetro é impresso no monitor serial. Então o tempo que o Led fica apagado e aceso é mostrado em milissegundos no monitor. As quatro últimas linhas são análogas às da programação do pisca com Led feita no terceiro encontro, só que com o tempo de `delay` sendo dado pelo valor do sensor.

Atividade 2: Após a montagem e programação, escrever no quadro o algoritmo 1 e orientar os estudantes a seguir os passos do algoritmo para abrir o monitor serial.

Algoritmo 1

- 1- Abra o Arduino IDE;
- 2- Clique em Tools, ou Ferramentas;
- 3- Clique em Serial monitor, ou monitor serial.
- 4- O monitor serial será aberto e, por fim, você verá as informações nele impressas.

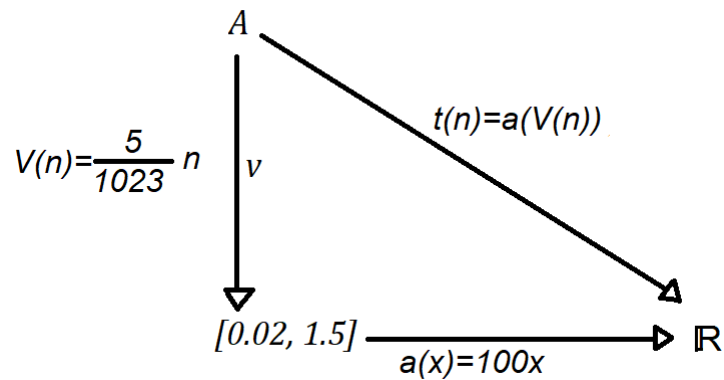


Figura 31 – Fonte: Autor. Feito com auxílio do software Paint.

Função discreta da porta analógica

A porta analógica do Arduino Uno serve apenas como entrada, e os valores captados por esta porta vão de $0000000000 = 0$ a $1111111111 = 1023$. No caso do potenciômetro utilizado nesta atividade, as funções definidas nas linhas 11 e 15 transformam o valor do sensor em um intervalo em milissegundos entre um ligar e um desligar do led. Desta forma, controlamos o intervalo do pisca utilizando a entrada analógica.

O tempo t em milissegundos de intervalo entre o ligar e o desligar do led é dado em função do sinal n analógico, em que

$$t(n) = n$$

5.2.6 Sexto encontro: Sensor de temperatura

Atividade 1: Explicar aos alunos como a função de conversão do sinal analógico em temperatura do sensor LM35 funciona, como descrito no texto explicativo a seguir:

Função de conversão do sinal analógico em temperatura do sensor LM35

O sensor LM35 pode medir temperaturas de -55°C a 150°C . Para tal, existem dois modos de ligar o sensor: A *ligação básica* e a *ligação de faixa completa*. A montagem que será feita neste encontro seguirá a ligação básica. Esta ligação mede temperaturas de 2°C a 150°C e, portanto, não pode medir as temperaturas abaixo de 2°C . Tais medidas só podem ser efetuadas na ligação de faixa completa, que mede de -55°C a 150°C .

A fórmula que transforma o sinal analógico n , dado pelo Arduino em função da tensão emitida pelo sensor, na temperatura t correspondente, em graus Celsius, é $t(n) =$

$$\frac{500n}{1023}$$

Para obter esta fórmula, utilizamos a função $a : [0.02, 1.5] \rightarrow \mathbb{R}$ na qual $a(x) = 100x$ que dá a temperatura a em graus Celsius em função da tensão x emitida pelo sensor, pois cada $0.01V$ vindo do sensor corresponde a $1^\circ C$, e esta tensão varia dentro do intervalo $[0.02, 1.5]$. Como o sensor é ligado a uma porta analógica, então as tensões que vão de $0.02V$ a $1.5V$, o que é apenas uma parte da tensão admitida pelas portas do Arduino ($0V$ a $5V$), são interpretadas como sinais analógicos pertencentes ao conjunto $\{5,6,7,\dots,306\}$, que também corresponde a uma parte equivalente dos sinais analógicos possíveis descritos anteriormente. Para obter a tensão correspondente ao sinal analógico dado, basta multiplicar este sinal pela constante de proporcionalidade $\frac{5}{1023}$. Daí, a função que dá a temperatura t em função do sinal analógico n é $t : A' \rightarrow \mathbb{R}$, onde $A' = \{5, 6, 7, \dots, 306\}$ e $t(n) = \frac{5}{1023} \cdot n \cdot 100 \Rightarrow C(n) = \frac{500n}{1023}$.

A equação utilizada na programação que será feita na atividade 1 está na forma padrão, como é orientado no manual e na apostila Robótica Livre com Arduino.

Atividade 1: Entregar a cada grupo os componentes do esquema de montagem na figura 30 e propor que os estudantes façam a montagem e a programação contidas nas figuras 30 e 31.

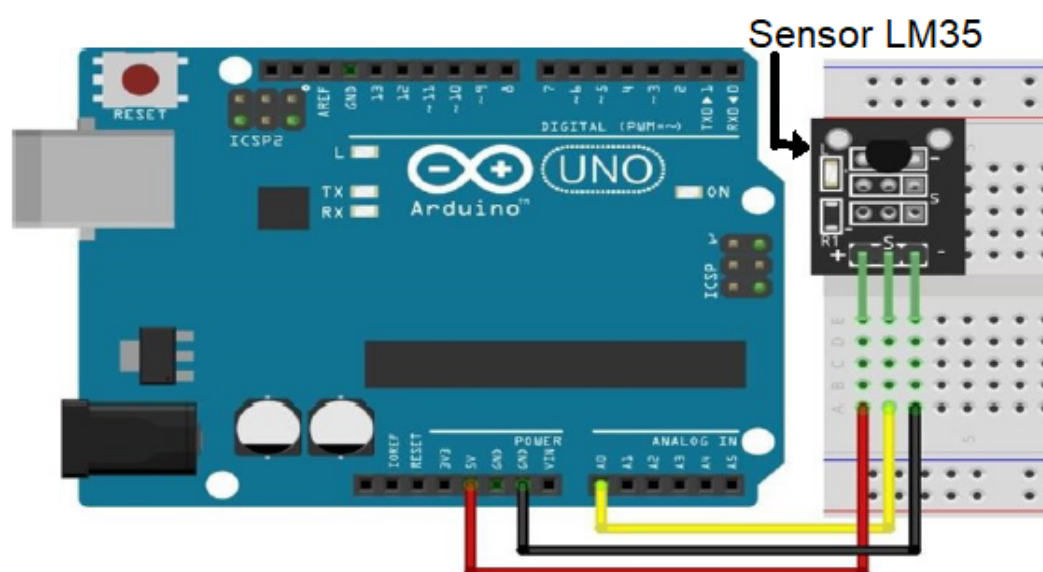


Figura 32 – Fonte: Autor. Adaptada da apostila robótica livre com Arduino

```
1  const int LM35 = A0; // Define o pino que vai ler a saída do LM35
2  float temperatura; // Variável que armazenará a temperatura medida
3  void setup() {
4  Serial.begin(9600);}
5  void loop() {
6  temperatura = (float(analogRead(LM35))*5/(1023))/0.01;
7  //Fórmula que transforma tensão em temperatura em °C
8  Serial.print("Temperatura (Celsius): ");
9  //Imprime "Temperatura (Celsius):" no monitor serial
10 Serial.println(temperatura);
11 //Imprime o valor de temperatura no monitor serial
12 delay(2000);}
```

Figura 33 – Fonte: Autor. Adaptação da apostila robótica livre com Arduino

Na parte de declaração de variáveis, o código declara que A0 é a porta que vai receber os valores do sensor LM35 e a variável que armazenará esses valores será “temperatura”. Na configuração, apenas o monitor serial é ativado com o comando “Serial.begin(9600)”. Na lógica, a variável temperatura é dada pela equação

$$temperatura = \frac{float(analogRead(LM35)).5}{1023} : 0,01$$

Esta equação transforma o valor do sensor LM35 lido pela porta A0 em uma temperatura na escala celsius. E, atribui esse valor à variável temperatura, que será impressa no monitor serial pelos comandos “serial.print”. A leitura é feita a cada 2 segundos, intervalo que pode ser alterado na linha delay(2000).

Atividade 2: Exibir a lei de correspondência da função t da atividade 1 e propor aos estudantes o problema 5.13 e 5.14

Problema 5.13. *Substitua a equação da linha 6 do código pela lei de correspondência da função t apresentada pelo professor e carregue o código para o Arduino. A temperatura continuou sendo medida corretamente? Explique este fato.*

Problema 5.14. *Pesquise as equações de conversão das escalas Kelvin e Fahrenheit para a escala Celsius e vice versa. Escreva um código para que a temperatura seja dada em Kelvin e depois faça o mesmo para a escala Fahrenheit.*

Nesta atividade, observamos o emprego das funções na resolução de problemas, que faz parte das habilidades EM13MAT302 e EM13MAT501, da BNCC, dispostas no início deste capítulo.

5.2.7 Sétimo encontro: Sensor de luminosidade e sensor ultrassônico

O sensor de luminosidade disponível no kit Arduino é um resistor dependente de luz ou, simplesmente, LDR que é um resistor que modula sua resistência em função da luz nele emitida. Quanto mais forte a luz, maior é a resistência que ele aplica ao circuito. Desta forma, ele se torna eficiente para dispositivos de relé e automações que sejam ativadas ou desativadas de acordo com a luminosidade ambiente.

Atividade 1: Entregar a cada grupo os componentes dispostos na lista a seguir e propor a montagem e a programação demonstradas nas figuras 32 e 33.

Componentes

- 1- Um sensor LDR;
- 2- Uma protoboard;
- 3- Uma placa Arduino Uno;
- 4- Um cabo USB de compilação do kit Arduino;
- 5- Três jumpers de cores distintas.

A montagem deve seguir o padrão na figura

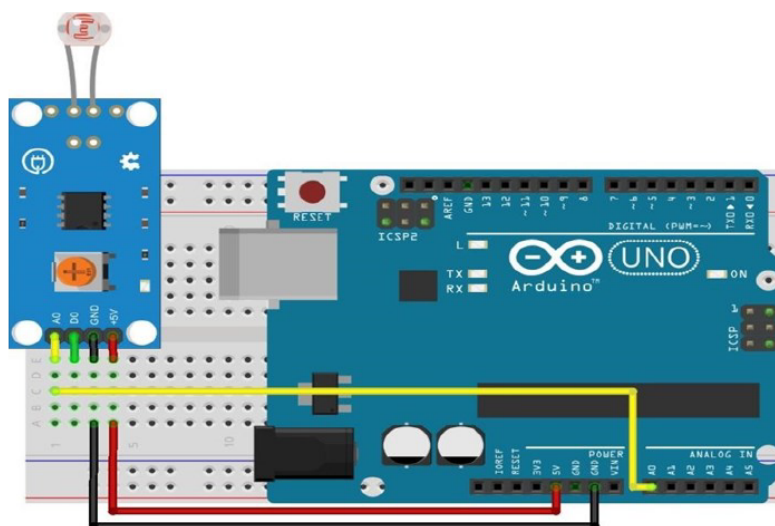


Figura 34 – Fonte: Apostila Robótica livre com Arduino, p. 16.

Observação: Os sensores LDR podem variar, dependendo do fabricante. Alguns mais atuais são menores, contendo apenas o disco espelhado e dois conectores, então a montagem pode precisar alterações.

Programação do LDR

O código sugerido é o seguinte:

```
1  int LDR = A0; //Declara a variável LDR cujos valores são recebidos
2      pela porta analógica A0.
3  int val = 0; //Declara a variável val, que será o valor analógico recebido do LDR.
4  void setup() {
5      Serial.begin (9600); //Inicia o monitor serial.
6  }
7  void loop() {
8      val = analogRead (LDR); //Adquire o valor na porta analogica e atribui a "val".
9      Serial.print (val); //Exibe no monitor serial o valor da variável "val".
10     delay (1000); //Espera 1000 mS = 1 segundo.
11 }
```

Figura 35 – Fonte: Autor. Adaptado da apostila Robótica livre com Arduino.

Atividade 2: Após a montagem e a programação, propor que os estudantes abram o monitor serial e, com as luzes da sala apagadas, anotem o valor impresso. Em seguida, acender as luzes da sala e pedir que os estudantes anotem o valor dado no monitor serial. Assim que cada grupo tiver os valores registrados, o professor deve propor que comparem os valores e discutam o porquê, caso haja diferença nos valores de um grupo e de outro.

Atividade 3: Distribuir a cada grupo os componentes listados a seguir e, na sequência, propor que os estudantes montem e programem o Sensor ultrassônico HC SR04, cujos esquemas estão dispostos nas figuras 34 e 35.

Componentes

- 1- Uma protoboard;
- 2- Uma placa Arduino Uno
- 3- Um cabo USB de compilação do kit Arduino;
- 4- Três pares de jumpers, sendo cada par com dois jumpers de cores iguais;
- 5- Um sensor ultrassônico HC SR04.

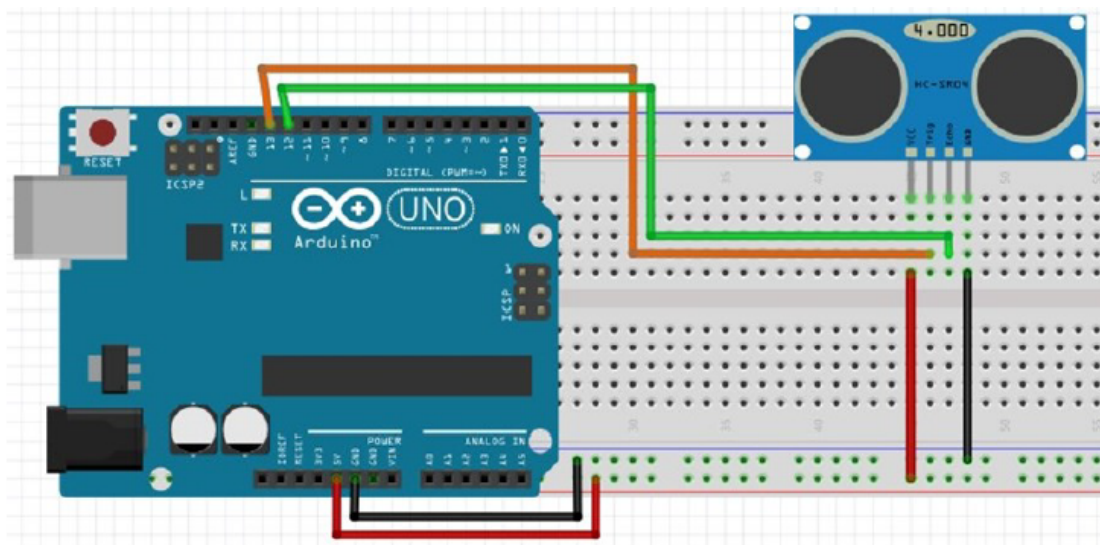


Figura 36 – Fonte: apostila Robótica livre com Arduino, p. 17.

Programação

O código para programação do sensor é

```

1  int trigPin = 13; //Declaração da porta de saída de emissão do pulso
2  int echoPin = 12; //Declaração da porta de recepção do pulso
3  void setup() {
4  Serial.begin (9600); //Ativa o monitor serial
5  pinMode (trigPin, OUTPUT); //Rotula a variável trigPin como saída de dados (emissão do pulso)
6  pinMode (echoPin, INPUT); //Rotula a variável echoPin como entrada de dados (recepção do pulso)
7  }
8  void loop() {
9  long duration, distance; //Declara as variáveis duration e distance
10 digitalWrite (echoPin, LOW); //Inicia o receptor de sinal como desligado
11 delayMicroseconds (2); //Aguarda 2 microssegundos
12 digitalWrite (trigPin, HIGH); //Aciona o emissor de sinal
13 delayMicroseconds (10); //Aguarda 10 microssegundos
14 digitalWrite (trigPin, LOW); //Desliga o emissor de sinal
15 duration = pulseIn (echoPin, HIGH); //contabiliza o instante entre a emissão e recepção do sinal sonoro
16 distance = (duration / 2) / 29.4; //Distância dada em função do tempo entre a emissão e a recepção do sinal
17 }

```

Figura 37 – Fonte: Descrições: Autor. Código: Autor, adaptado da apostila Robótica livre com Arduino, p. 18.

$$\textit{Explicação: } d = vt = \frac{340m}{s} \cdot \frac{duration}{2} = 340 \cdot \frac{100cm}{10^6\mu s} \cdot \frac{duration}{2} \cdot 0.034 \approx \frac{1}{29,4}$$

Nesta atividade, será utilizado um sensor ultrassônico HC SR04 e, utilizando a propriedade física da reflexão das ondas, este sensor servirá para medir a distância entre ele e um objeto à frente. O HC SR04 emite um pulso ultrassônico e também o recebe. A diferença entre o instante da emissão e o recebimento é um intervalo de tempo que

será transformado em distância através da função horária da posição.

Atividade 4: Resolver o problema 5.16 no quadro junto com os estudantes

Problema 5.15. *A velocidade do pulso ultrassônico da HC SR04 é a mesma do som no ar, que é de, aproximadamente, 340 m/s. Utilizando o cálculo da distância d de um movimento, em função da velocidade v , em metros por segundo e do tempo t em segundos, que é*

$$d = v \cdot t,$$

determine a lei de correspondência da função d , que dá a distância do sensor ao objeto imediatamente à sua frente, a partir do tempo t , em segundos.

Solução

O sensor calcula o tempo em microssegundos que o pulso ultrassônico leva do momento em que sai do sensor até o momento em que retorna. Desta forma, o tempo t em microssegundos calculado pelo sensor deve ser dividido por dois, pois é o tempo de ida e volta do pulso, já que ele reflete no objeto. O tempo obtido na função `pulseIn` é medido em microssegundos, que equivalem a 10^{-6} vezes um segundo. (0,000001 s) [3].

Assim, a distância d , em metros, em função do tempo obtido pelo sensor pode ser escrita como

$$d(t) = \frac{t}{2} \cdot 10^{-6} \cdot 340 \iff d(t) = 17 \cdot 10^{-5}t$$

Como a linguagem matemática utilizada no Arduino se baseia apenas nas quatro operações básicas, a função encontrada acima pode ser escrita como

$$d(t) = 0,00017t$$

Atividade 5: Propor que os(as) estudantes resolvam os problemas 5.17 e 5.18

Problema 5.16. *Realize medições de distância com seu sensor na sala de aula e conjecture qual a unidade de medida na qual a distância dada pelo HC SR04 é mostrada no monitor serial.*

Problema 5.17. *2- Substitua a equação da linha 16 do código pela que o professor mostrou na resolução do problema 6.16. A unidade na qual a distância é mostrada é realmente metros?*

5.2.8 Oitavo encontro: Buzzer e sensor de campo magnético

Esta atividade consistirá em programar uma saída sonora, que é o buzzer. Para isto, utilizaremos a função `tone`, a qual pode reproduzir músicas e melodias por meio do buzzer a partir das frequências das notas musicais.

Atividade 1: Entregar os componentes listados a seguir para cada grupo e propor que montem e programem a saída sonora buzzer, cujos esquemas de montagem e programação constam nas figuras 36 e 37.

Componentes

- 1- Uma placa protoboard;
- 2- Uma placa Arduino Uno;
- 3- Um cabo USB de compilação do kit Arduino;
- 4- Um buzzer;
- 5- Dois jumpers de cores distintas.

Dependendo do modelo do buzzer, a montagem abaixo pode necessitar de adaptações. Existe uma variação de montagem com a qual é possível fazer o som do buzzer ser emitido em um volume mais alto. Não é difícil, por experimentação, encontrar esta montagem. Para isto, o professor pode disponibilizar um jumper a mais para os grupos.

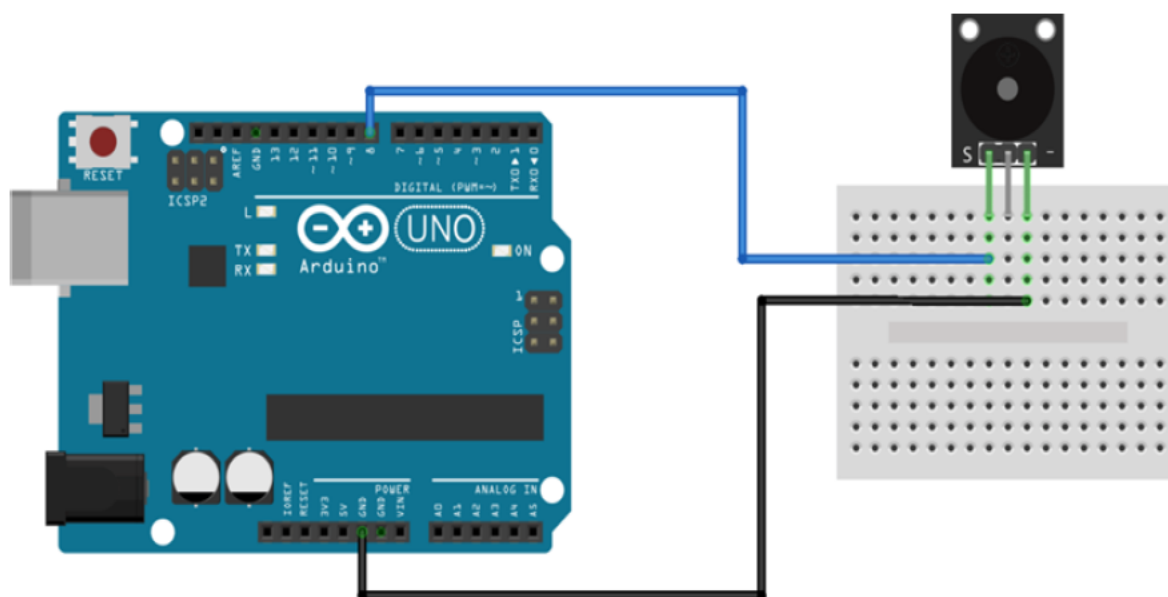


Figura 38 – Fonte: apostila Robótica livre com Arduino, p. 23.

Programação

```
1  #define tempo 10
2  int frequencia = 0;
3  int Pinofalante = 8;
4  void setup() {
5  |   Serial.begin(115200);
6  pinMode(Pinofalante,OUTPUT); //Pino do buzzer
7  }
8  void loop()
9  {
10 |   //Varia frequência entre 150 e 1800 somando 5
11 for (frequencia = 150; frequencia < 1800; frequencia += 5)
12 {
13 tone(Pinofalante, frequencia, tempo);
14 Serial.println(frequencia);
15 delay(1);
16 }
17 //Varia frequencia entre 1800 e 150 subtraindo 5
18 for (frequencia=1800; frequencia >150; frequencia-=5)
19 {
20 |   tone(Pinofalante, frequencia, tempo);
21   Serial.println(frequencia);
22   delay(1);
23 }
24 }
```

Figura 39 – Fonte: Autor. Adaptado a partir da Apostila Robótica livre com Arduino.

Uma pergunta que pode surgir é: Como é possível oscilar a frequência em mais de 2 tons diferentes se o buzzer está ligado em uma porta digital, que só recebe dois bytes?

A resposta a essa pergunta é que o tom do som é controlado por uma frequência de pulsos de onda, e não por tensões específicas. Assim, o Arduino pode usar a porta digital para emitir pulsos com a mesma tensão e mudar apenas a frequência dos pulsos, o que alterará o tom emitido pelo buzzer.

Atividade 2: Propor o problema 5.19

Problema 5.18. *Seja $t \geq 0, t \in \mathbb{N} \cup \{0\}$, o instante em milissegundos em que a frequência $f(t)$ é emitida pelo buzzer. Pela programação, sabemos que no instante 0, a frequência é 150 Hz e aumenta 5Hz a cada 1 milissegundo até chegar a 1800 Hz, que é quando esta frequência começa a diminuir 5 Hz a cada segundo. A partir disso, responda:*

- (a) Quanto vale $f(10)$? E $f(25)$?
- (b) Em que momento t a frequência começa a diminuir?

(c) Sendo $A = \{t; 0 \leq t \leq 330\}$ o domínio da função f , determine a lei de correspondência de f .

(d) Determine a lei de correspondência de f se o domínio é $D = \{t \in \mathbb{N}; 331 \leq t \leq 660\}$.

Atividade 3: Entregar a cada grupo os componentes listados a seguir e pedir que montem e programem o sensor de campo magnético cujo esquema de montagem e programação consta nas figuras 38 e 39.

Componentes

Para esta atividade, cada grupo precisará de

- 1- Uma placa protoboard;
- 2- Uma placa Arduino Uno;
- 3- Um cabo USB de compilação do kit Arduino;
- 4- Um sensor de efeito hall 49e;
- 5 - Três jumpers de cores distintas;
- 6- Um ímã qualquer.

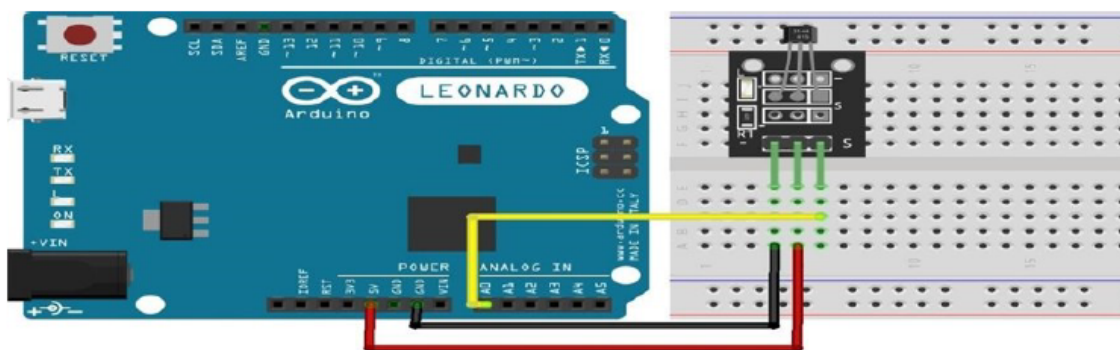


Figura 40 – Fonte: apostila Robótica livre com Arduino, p. 20.

Programação

```

1 float refVoltage = 5.0/1023; //Fixa a constante referência
2 float sensorVolts;          //Cria a variável da tensão vinda do sensor
3 int val1;                   //Cria a variável do valor analógico de A0
4 void setup() {
5   Serial.begin(9600);       //Inicia comunicação com o monitor serial
6 }
7 void loop() {
8   val1=analogRead(0); //Lê o valor da porta A0 e o atribui à variável val1
9   sensorVolts=refVoltage*(val1);
10  // Multiplica o sinal de A0 por 5/1023 calculando assim a tensão
11  Serial.print("B=");
12  //Imprime "B" no serial, como a variável do campo magnético
13  Serial.print((sensorVolts*895.52773449)-2238.8193362 );
14  // Multiplica a tensão por uma constante e subtrai outra, obtendo o valor de B
15  Serial.println("Gauss");//Imprime a palavra "Gauss" no serial
16  if (((sensorVolts*667)-1667) > 2) { Serial.println (" Norte");};
17  if (((sensorVolts*667)-1667) < -2 ) {Serial.println (" Sul");};}
18  //Define se o campo magnético é norte ou sul pela tensão tensão vinda do sensor.

```

Figura 41 – Fonte: Autor. Adaptado a partir da apostila Robótica livre com Arduino, ps. 20 e 21.

Nesta atividade, é feita a programação e montagem de um sensor de campo magnético que identifica a polaridade de um ímã. Porém, o sensor hall49e pode ser utilizado para programar diversos dispositivos diferentes envolvendo campos magnéticos, como detectores de metal, por exemplo. Quando o hall 49e é atravessado por um campo magnético, ele produz um corrente elétrica proporcional ao campo, que vai de 0,86V a 4,21 V e mede campos elétricos de -1500 Gauss a +1500 Gauss.

Atividade 4: Propor os problemas 5.18 e 5.19

Problema 5.19. Na equação da linha 9, *sensorVolts* é a tensão obtida por multiplicar *refVoltage* por *val1*, onde a variável *val1* é o valor analógico recebido pela porta A0 e *refVoltage* é a constante $\frac{5}{1023}$. Baseando-se nesta equação, responda às questões a seguir:

- Seja t a tensão, e n o valor analógico, escreva a lei de formação da função $t(n)$.
- Determine os valores máximo e mínimo da função t

Problema 5.20. A expressão na linha 13 do código calcula o valor do campo magnético captado pelo sensor em função da tensão advinda dele, representada por *sensorVolts*.

Seja $c(t)$ a função definida pela expressão da linha 13 do código, que determina o campo magnético a partir da tensão t vinda do sensor.

- Usando o item (c) da questão 1, determine o domínio da função c .

- (b) *Escreva a lei de formação da função c usando a expressão da linha 13 do código.*
- (c) *Calcule $c(2)$, $c(3)$ e $c(5)$.*
- (d) *Calcule os valores máximo e mínimo da função c .*
- (e) *Qual o tipo da função (c) ?*

5.2.9 Análise da aplicação

Faremos uma breve análise sobre as aplicações realizadas neste trabalho, refletindo sobre quais dos objetivos para com o currículo e o aprendizado podem ser alcançados e que ajustes podem ser feitos para melhor contemplar as propostas da disciplina através desta sequência.

As aplicações deste trabalho ocorreram nos anos de 2023 e 2024, na escola de referência em Ensino Médio Padre Zuzinha em quatro turmas diferentes, formadas por estudantes que optaram pela eletiva de robótica.

A proposta para as eletivas era que elas contemplassem alguma das áreas de conhecimento. Assim, a eletiva de robótica visava contemplar conhecimentos das áreas de matemática e física.

Nas atividades de programação e montagem, as turmas eram divididas em grupos de quatro integrantes, conforme proposto no início deste capítulo. Na resolução dos problemas matemáticos, cada estudante fez de maneira individual, no próprio caderno ou no material impresso.

As turmas que participaram da eletiva foram turmas do segundo ano do Ensino Médio. Assim, era pressuposto que eles já conheciam o conceito de função. Na aplicação das atividades, era notável que todos os estudantes apresentavam disposição para participar, visto que as atividades de montagem e manipulação dos dispositivos são um atrativo para os jovens. Além disso, programar o protótipo montado de várias maneiras distintas se tornava uma prática divertida para eles, o que motivou bastante a participação de todos nas atividades.

Por ter sido uma eletiva de robótica, e a avaliação das eletivas não tinha caráter reprobatório, alguns(mas) estudantes não foram assíduos com as resoluções dos problemas como deveriam. Um ponto a ser considerado é que as atividades de resolver problemas valiam uma certa nota para a avaliação somativa.

Ao fim da aplicação da eletiva, foi proposto que cada grupo desenvolvesse um projeto próprio, visando criar uma solução que auxiliasse as pessoas a enfrentar um problema real e apresentassem à turma, atividade a qual não foi realizada por todos os estudantes. A boa parte dos grupos optou por não realizar tal projeto ou apresentaram apenas o esboço de um projeto, não pondo em prática sua programação e montagem.

Em 2023, um grupo desenvolveu um protótipo de um sensor de ré com Arduino, do qual não dispomos de registro. No ano seguinte, o mesmo grupo desenvolveu um

dispositivo de ventilação para uma caixa de energia do Moda Center, em Santa Cruz do Capibaribe. Um dos integrantes do grupo que à época trabalhava no Moda Center, e observara que a caixa de energia estava apresentando falhas no funcionamento em dias de muito calor, o que é típico da cidade. A partir daí surgiu a ideia de criar uma ventoinha acoplada a um sensor de temperatura utilizando o material do Arduino. Na figura 37 segue uma foto do protótipo.

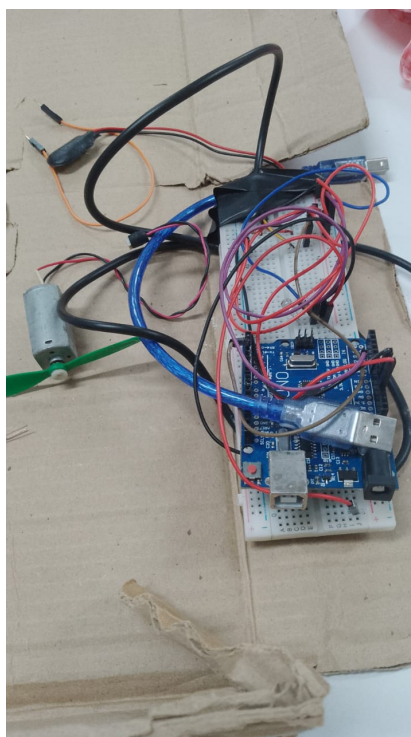


Figura 42 – Fonte: autor, foto cedida por integrante do grupo desenvolvedor.

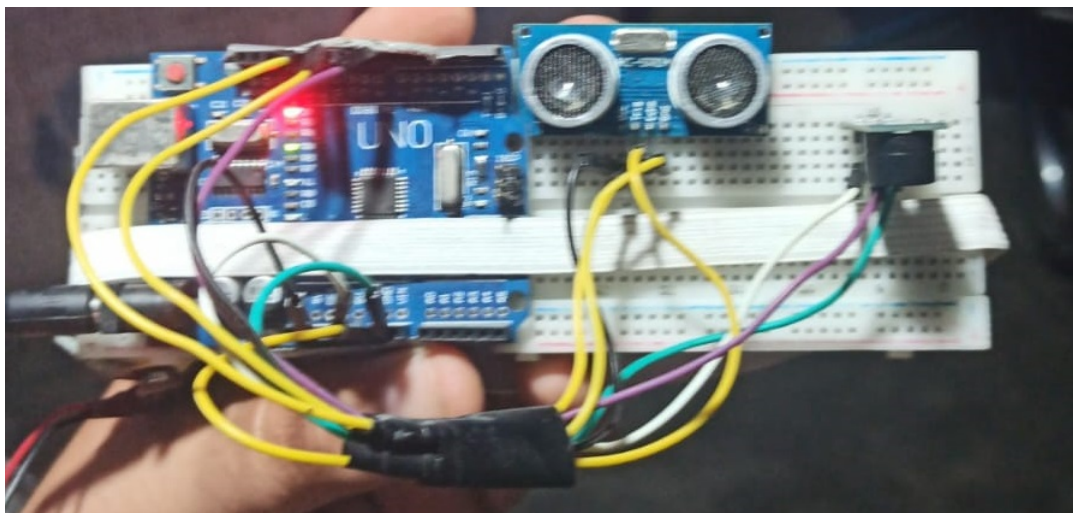


Figura 44 – Fonte: Foto cedida por estudante



Figura 43 – Fonte: autor, foto cedida por integrante do grupo desenvolvedor.

Em 2024, este trabalho foi apresentado na 1ª mostra de robótica da Gerência Regional de Educação do agreste setentrional de Pernambuco, que aconteceu no Armazém da Criatividade, no polo comercial de Caruaru - PE. Neste evento, os estudantes puderam apresentar sua ideia a estudantes de outras escolas, visitantes e empresários do ramo da tecnologia, dos quais alguns demonstraram interesse no projeto e, ofereceram propostas de cursos e trabalhos futuros.

Um outro grupo, da turma de 2024, desenvolveu um sensor para óculos destinado a deficientes visuais. O protótipo consistia em uma sutil saída de som que é ativada quando um objeto se aproxima a menos de um metro e meio à frente do rosto de quem está utilizando. A proposta era que este protótipo fosse diminuído e acoplado a um par de óculos, se tornando mais discreto e elegante. Outra ideia seria incluir uma vibração para servir como alternativa ao sinal sonoro.

6 Conclusão

A robótica e a programação atreladas ao ensino de matemática se tornam meios para que o estudante possa aplicar seus conhecimentos matemáticos na resolução de problemas reais, além de serem conhecimentos que estimulam a criatividade.

Em 2022, foi realizado pela OCDE a avaliação PISA (*Programme for International Student Assessment* - Programa Internacional de Avaliação de Estudantes), na qual foram avaliados estudantes de 64 países, dentre eles, o Brasil. Os alunos foram avaliados em três áreas: Leitura, Matemática e Ciências. Na área de matemática, é avaliada a capacidade do aluno usar conceitos matemáticos para resolver problemas do mundo real e criar soluções a partir de seus conhecimentos.

Em 2024, a OCDE publicou o ranking com os resultados dos países em cada área. O Brasil teve um resultado alarmantemente baixo no que diz respeito à capacidade de resolver problemas e na inventividade. Obteve a 49^a colocação, com uma pontuação de 23 pontos, 10 pontos abaixo da média do PISA, que era 33 pontos. Com isto, é notável a necessidade de incentivar os estudantes a desenvolver o pensamento criativo e a resolução de problemas.

A habilidade *EM13MAT405* da BNCC propõe que o aluno aprenda a utilizar algoritmos de programação para resolver problemas. Neste sentido, desenvolvemos este trabalho, que tem, por tema, a aplicação de funções e algoritmos de programação na robótica e na automação.

Nos primeiros capítulos do trabalho, mostramos a história das automações. Iniciamos pela antiguidade, passando pela revolução industrial e chegando finalmente aos dias atuais. Em seguida, os conhecimentos de matemática mais importantes para a aplicação da sequência didática proposta, incluindo o conceito de conjunto, os algoritmos e as funções. Passamos então aos algoritmos de programação e à plataforma Arduino, incluindo sua história e a causa de sua popularização na atualidade. Por fim, apresentamos as atividades desenvolvidas na escola, nas quais os temas funções e algoritmos de programação são utilizados nas atividades de montagem e programação de cada dispositivo.

As atividades foram designadas de forma que estimule o(a) estudante a explorar a versatilidade da plataforma Arduino, de maneira que ele(a) a possa enxergar como uma ferramenta para a criatividade e resolução de problemas reais.

Durante a sequência das atividades com Arduino, o(a) estudante tem problemas para solucionar nos códigos de programação, nos quais ele precisa aplicar os conhecimentos sobre funções e algoritmos aprendidos em sala de aula, o que possibilita uma aprendizagem mais envolvente desses conteúdos, sendo acompanhada pela montagem

dos dispositivos robóticos.

Neste trabalho, oferecemos ao professor de matemática do Ensino Básico, uma alternativa para a aplicação desses temas, de modo que ele, junto com os(as) estudantes, possam sair da rotina de sala de aula e dê a oportunidade de aprender os conteúdos utilizando algo diferente, e que costuma gerar interesse nos(as) estudantes. Além disso, o professor pode exemplificar, para o(a) estudante, como os conhecimentos matemáticos estão presentes na área de tecnologia e no nosso dia a dia, quando utilizamos qualquer aparelho eletrônico, desde ventiladores até os smartphones, o que aproxima a matemática da visão de mundo do(a) estudante.

Almejamos que este trabalho possa vir a ser utilizado por muitos professores e que possa se tornar um recurso didático para suas escolas.

Referências

Nenhuma citação no texto.

V. V. Amelkin. *Differential Equations in Applications*. CRC Press, Boca Raton, 1999. Citado 2 vezes nas páginas [15](#) e [16](#).

Arduino Documentation. `pulsein()` — referência de linguagem arduino, 2024. Acessado em 27 de maio de 2025. Citado na página [64](#).

TF Bell. *Jacquard weaving and designing*. Longmans, Green, 1895. Citado na página [18](#).

BRASIL. Base nacional comum curricular, 2017. BNCC. Citado 2 vezes nas páginas [35](#) e [43](#).

SET Comunicação. Inteligência artificial ajuda a salvar vidas no coração da amazônia. *SET Comunicação*, 7 2025. Acesso em 17 jul. 2025. Citado na página [21](#).

Renato Cunha. O tear jacquard não só revolucionou a indústria têxtil mas foi o primeiro computador do mundo, 2017. Citado 2 vezes nas páginas [18](#) e [19](#).

Wellington Gonçalves DE ANDRADE. A máquina de heron: Desenvolvimento de um procedimento experimental para o ensino da termodinâmica a fim de entender as máquinas térmicas. *Revista Semiárido De Visu*, 2021. Citado na página [18](#).

José Altenis dos Santos and Gabriel Pimenta Carneiro Campelo. Robótica livre com arduino. Apostila, 2022. Disponível em: <https://pt.scribd.com/document/567775122/APOSTILA-Robotica-Livre-com-Arduino>. Citado na página [43](#).

Mouser electronics. A história do arduino — parte 1: Apresentando o arduino, 2025. Citado na página [36](#).

Redação Galileu. Inteligência artificial ajuda arqueólogos a traduzir textos de milênios. *Revista Galileu*, 2023. Acesso em 17 jul. 2025. Citado na página [21](#).

Eric J Hobsbawm. *A era das revoluções: 1789-1848*. Editora Paz e Terra, 2012. Citado na página [18](#).

Benigno José Bayán Jardín. Construcción de un reloj de agua o clepsidra. *Dyna*, 82(8):423–430, 2007. Citado na página [16](#).

David Kushner. The making of arduino: How five friends engineered a small circuit board that's taking the diy world by storm, 2011. Citado na página [36](#).

Justin Lahart. Taking an open-source approach to hardware. *The Wall Street Journal*, 2009. Citado na página [36](#).

Elon Lages Lima. *Análise Real: Volume 1*. Instituto Nacional de Matemática Pura e Aplicada (IMPA), Rio de Janeiro, 8 edition, 2009. Citado na página [28](#).

Elon Lages Lima. *Números e Funções Reais*. Number 7 in Coleção PROFMAT. Sociedade Brasileira de Matemática (SBM), Rio de Janeiro, RJ, 2 edition, 2013. Citado na página 25.

Martins, Roberto de Andrade. As máquinas simples na “mecânica” de heron de alexandria. *História da ciência e ensino: fontes primárias*, 2:15–30, 2018. Citado na página 17.

Luiz Miranda. Prototipagem: o que é, como funciona e para o que serve. *Revista Quero bolsa*, December 2023. Citado na página 12.

Ouvidorias.gov.br. Design thinking, 2023. Acesso em: 22 de outubro de 2023. Citado na página 12.